

Western  Graduate&PostdoctoralStudies

Western University  
**Scholarship@Western**

---

Electronic Thesis and Dissertation Repository

---

10-20-2016 12:00 AM

## Computation of Real Radical Ideals by Semidefinite Programming and Iterative Methods

Fei Wang

*The University of Western Ontario*

Supervisor

Greg Reid


*The University of Western Ontario*

Graduate Program in Applied Mathematics

A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy

© Fei Wang 2016

Follow this and additional works at: <https://ir.lib.uwo.ca/etd>

 Part of the [Algebraic Geometry Commons](#), and the [Other Applied Mathematics Commons](#)

---

### Recommended Citation

Wang, Fei, "Computation of Real Radical Ideals by Semidefinite Programming and Iterative Methods" (2016). *Electronic Thesis and Dissertation Repository*. 4262.

<https://ir.lib.uwo.ca/etd/4262>

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact [wlsadmin@uwo.ca](mailto:wlsadmin@uwo.ca).

# Abstract

Systems of polynomial equations with approximate real coefficients arise frequently as models in applications in science and engineering. In the case of a system with finitely many real solutions (the 0 dimensional case), an equivalent system generates the so-called real radical ideal of the system. In this case the equivalent real radical system has only real (i.e., no non-real) roots and no multiple roots. Such systems have obvious advantages in applications, including not having to deal with a potentially large number of non-physical complex roots, or with the ill-conditioning associated with roots with multiplicity. There is a corresponding, but more involved, description of the real radical for systems with real manifolds of solutions (the positive dimensional case) with corresponding advantages in applications.

The stable and practical computation of real radicals in the approximate case is an important open problem. Theoretical advances and corresponding implemented algorithms are made for this problem.

The approach of the thesis is to use semidefinite programming (SDP) methods from algebraic geometry and also techniques originating in the geometry of differential equations. The problem of finding the real radical is re-formulated as an SDP problem. This approach in the 0 dimensional case was pioneered by Curto & Fialkow with breakthroughs in the 0 dimensional case by Lasserre and collaborators. In the positive dimensional case, important contributions have been made of Ma, Wang and Zhi. The real radical corresponds to a generic point lying on the intersection of boundary of the convex cone of positive semidefinite matrices and a linear affine space associated with the polynomial system.

As posed, this problem is not stable, since an arbitrarily small perturbation takes the point to an infeasible one outside the cone. A contribution of the thesis is to show how to apply facial reduction pioneered by Borwein and Wolkowicz to this problem. It is regularized by mapping the point to one which is strictly on the interior of another convex region, the minimal face of the cone. Then a strictly feasible point on the minimal face can be computed by accurate iterative methods such as the Douglas-Rachford method. Such a point corresponds to a generic point (max rank solution) of the SDP feasible problem. The regularization is done by solving the auxiliary problem which can be done again by iterative methods. This process is proved to be stable under some assumptions in this thesis as the max rank doesn't change under sufficiently small perturbations. This well-posedness is also reflected in our examples, which are executed much more accurately than by methods based on interior point approaches.

For a given polynomial system, and an integer  $d > 0$ , results of Curto & Fialkow and Lasserre are generalized to give an algorithm for computing the real radical up to degree  $d$ . Using this truncated real radical as input to critical point methods can lead in many cases to validation of the real radical.

**Keywords:** SDP Optimization, Numerical Algebraic Geometry, Facial Reduction

## Co-Authorship Statement

Chapters 2 -4 of this thesis consist of the following papers:

Chapter 2: Greg Reid, Fei Wang and Wenyuan Wu. A note on geometric involutive bases for positive dimensional polynomial ideals and SDP methods. *Proceedings of the 2014 Symposium on Symbolic-Numeric Computation*, Pages 41-42. ACM, 2014.

Chapter 3: Greg Reid, Fei Wang, Henry Wolkowicz and Wenyuan Wu. Semidefinite Programming and facial reduction for Systems of Polynomial Equations. Submitted to *Theoretical Computer Science*, arXiv:1504.00931, 2015.

Chapter 4: Fei Wang, Greg Reid and Henry Wolkowicz. Finding Maximum Rank Moment Matrices by Facial Reduction and Douglas-Rachford Method. arXiv:1606.00491, 2016

The original draft for each of the above articles was prepared by the author and Greg Reid. Subsequent revisions were performed by the author and Dr. Greg Reid. Development of software, analytical and numerical work using MATLAB was performed by the author under supervision of Dr. Greg Reid.

## Acknowledgements

I would like to express my sincere gratitude to my advisor Prof. Greg Reid for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I really enjoy working with Greg who created an wonderful atmosphere which especially stimulates my ability to work independently. I could not have imagined having a better advisor and mentor for my Ph.D study. I also thank Greg for supporting me to attend many conferences and workshops where I can present my work and meet many experts in my field of study, for example, the wonderful to the SNC Shanghai conference to present my research. There are also many other examples and I can't list them all here.

Besides my advisor, I also owe a lot of gratitude to my supervisory committee members Rob Corless and David Jeffrey. Before I started working in the field of real algebraic geometry, I worked with them on the Lambert W function. One of the work is published in ISSAC 14 Japan. Without their support, this could not be possible. Although my work on Lambert W function is not the topic of this thesis, the experience gained by working with them has helped me a lot in many aspects of my research. I also thank Rob Corless and David Jeffrey for many workshops they organized to create an inspiring environment inside the department. For example, they organised the workshop "20 years of Lambert W function" where I presented my work and meet with people from many diverse and interesting research fields.

My sincere thanks goes to Prof. Henry Wolkowicz who is an expert in optimization. I learned a lot of things about optimization from Henry. Henry introduced me to the idea of facial reduction and Douglas-Rachford iteration which is very important in this thesis. I also thank him for his many useful comments and references he sent to me which greatly widens and deepens my knowledge about optimzation. My sincere thanks also goes to Wenyuan Wu who is one of the coauthors of the paper for Chapter 2 in this thesis. The discussion with him about the real radical and critical points gave lots of inspirations to me.

Last but not least, I would like to thank my family members and my girl friend for their support, patience and understanding throughout my PhD study which I can't describe in words.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Co-Authorship Statement</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>x</b>
<b>List of Appendices</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Real and complex solution sets (varieties) of systems of polynomial equations .	2
1.2 Equivalent systems of polynomials: generators of ideals and radicals of polynomial systems . . . . .	5
1.3 Introductory example of computation of the real radical using Moment Matrices and SDP . . . . .	9
1.4 SDP optimization . . . . .	10
1.4.1 Semidefinite Matrices . . . . .	10
1.4.2 Semidefinite Programs . . . . .	11
1.4.3 Face, minimal face and facial structure . . . . .	12
1.4.4 Facial reduction . . . . .	12
1.5 Moment problem . . . . .	13
1.5.1 Linear form, positive linear form and moment matrix . . . . .	13
1.5.2 Moment Problem . . . . .	15

1.5.3	Truncated Moment matrix and flat extension theorem . . . . .	15
1.5.4	Generic linear forms . . . . .	17
1.6	Outline of the contents of the thesis . . . . .	17
1.6.1	Contents of Chapter 2 . . . . .	18
1.6.2	Contents of Chapter 3 . . . . .	18
1.6.3	Contents of Chapter 4 . . . . .	19
1.6.4	Conclusions are given in Chapter 5 . . . . .	20
1.6.5	Appendices . . . . .	20
	Bibliography . . . . .	20

## 2 Geometric involutive bases for positive dimensional polynomial ideals and SDP

	<b>methods</b>	<b>24</b>
2.1	Introduction . . . . .	24
2.2	Brief background on ideals and varieties . . . . .	26
2.2.1	Some basic objects in complex algebraic geometry . . . . .	26
2.2.2	Some basic objects in real algebraic geometry . . . . .	28
2.3	Geometric prolongation and projection for polynomial systems . . . . .	29
2.4	Geometric involutive bases . . . . .	33
2.4.1	Symbol, class and Cartan involution test . . . . .	33
2.4.2	Projected involutive form algorithm . . . . .	36
2.5	Moment matrices and SDP . . . . .	38
2.5.1	Moment Matrices . . . . .	38
2.5.2	Moment matrix for univariate example . . . . .	38
2.6	Combining geometric involutive bases and moment matrix methods . . . . .	40
2.6.1	Geometric involutive form and moment matrix algorithms . . . . .	40
2.6.2	Two variable example . . . . .	42
2.6.3	Three variable example . . . . .	44
2.7	Discussion . . . . .	46
	Bibliography . . . . .	47

<b>3 Semidefinite Programming and facial reduction for Systems of Polynomial Equations</b>	<b>53</b>
3.1 Introduction . . . . .	53
3.2 Real radical ideals and moment matrices . . . . .	56
3.2.1 Real polynomial systems . . . . .	56
3.2.2 Moment matrices . . . . .	58
3.3 Geometric involutive bases . . . . .	59
3.4 Combining the moment matrix and geometric involutive form algorithms . . . .	63
3.5 Facial reduction and projection methods . . . . .	65
3.5.1 Representations for linear constraints for moment problems . . . . .	65
3.5.2 First step of facial reduction . . . . .	68
Potential second facial reduction . . . . .	69
Backward stability for facial reduction steps . . . . .	71
3.5.3 Projection methods . . . . .	72
Method of alternating projections, MAP . . . . .	72
Douglas-Rachford reflection method . . . . .	73
3.6 Numerical experiments . . . . .	75
3.6.1 A class of random univariate polynomials . . . . .	75
3.6.2 Examples of Ma, Wang and Zhi [37] . . . . .	77
3.6.3 Intersecting higher dimensional cylinders . . . . .	82
3.7 Conclusion . . . . .	83
Bibliography . . . . .	86
 <b>4 Maximum Rank Moment Matrices by Facial Reduction and Douglas-Rachford Method</b>	 <b>93</b>
4.1 Introduction . . . . .	93
4.2 Moment Matrices . . . . .	95
4.3 SDP and facial reduction . . . . .	96
4.3.1 Faces . . . . .	97
4.3.2 Theorems of the alternative . . . . .	97

4.3.3	Facial reduction . . . . .	98
4.3.4	Facial reduction maximum rank algorithm . . . . .	100
4.3.5	Transform of the auxiliary problem . . . . .	102
4.4	Projection method . . . . .	103
4.4.1	Projection to the positive semidefinite cone . . . . .	103
4.4.2	Projection to an affine subspace . . . . .	104
4.4.3	Douglas-Rachford method . . . . .	104
4.5	The ill-conditioned case . . . . .	105
4.6	Well-posedness . . . . .	106
4.7	Computation of generators of the real radical up to a given degree . . . . .	108
4.8	A special case for determining positive dimensional real radical . . . . .	113
4.9	Comparison with Triangular decomposition of semi-algebraic sets . . . . .	115
4.10	Examples . . . . .	115
4.11	Conclusion . . . . .	120
	Bibliography . . . . .	121
<b>5</b>	<b>Conclusion</b>	<b>128</b>
	Bibliography . . . . .	131
<b>A</b>	<b>Proof of Primal Theorem of Alternative</b>	<b>133</b>
<b>B</b>	<b>Copyright Release</b>	<b>135</b>
	<b>Curriculum Vitae</b>	<b>137</b>



# List of Figures

2.1	Table of $\dim \pi^\ell \mathbf{D}^k P$ for (2.17) for Example 2.3.2. The (red) boxed 4 in the first column corresponds to $\pi^4 P$ and a geometric involutive basis for $P$ as found by Algorithm 2.4.1. The blue and black boxed 4's in the fourth column correspond to geometric involutive bases for $P$ . . . . .	32
2.2	Table of $\dim \pi^\ell \mathbf{D}^k(P_2)$ for system (2.41) The (blue) boxed 11 in the third column corresponds to $\pi^2 \mathbf{D}^2(P_2)$ . . . . .	43
2.3	Table of $\dim \pi^\ell \mathbf{D}^k \text{gen}(\ker M)$ for the first GIF– M iteration in Example 2.6.2. . . . .	44
2.4	Table of $\dim \pi^\ell \mathbf{D}^k(P_3)$ for system (2.43). . . . .	44
2.5	Table of $\dim \pi^\ell \mathbf{D}^k \text{gen}(\ker M)$ in the moment matrix calculation for $\text{gen}(\pi^2 \mathbf{D}^0(P_3))$ . . . . .	45
3.1	Comparison in residual and cputime of GIF-FDR vs GIF-SeDuMi for random polynomials $p_d(x) = \sum_1^d a_{d,j} x^j$ at odd degrees $3 \leq d \leq 51$ with $a_{d,j} \sim N(0, 1)$ . . . . .	76
3.2	Performance profile of GIF-FDR vs GIF-SeDuMi for random polynomials $p_d(x) = \sum_1^d a_{d,j} x^j$ at each odd degrees $3 \leq d \leq 51$ with $a_{d,j} \sim N(0, 1)$ . The profile function used is (3.26). . . . .	77
4.1	In the Figure, the black monomial staircase represents the leading monomials of the generators of the real radical determined to degree $d$ by $\text{RealRadical}(F, d)$ . The only way these can fail to be a complete set of generators for the real radical is that there is a minimum degree $d' > d$ where additional generators with leading monomials of exactly degree $d'$ shown in red are found outside black monomial staircase. . . . .	113

# List of Tables

3.1	block partitioned bivariate moment matrix; submatrices have same degree . . .	66
3.2	<b>Statistics for the application of GIF-FDR and GIF-SeDuMi:</b> Ex 4.1-4.6 are 6 examples in MWZ [37]; Cyl2d-Cyl5d are cylinder examples; $n$ number of variables; $d$ maximum polynomial degree; $m$ number of polynomials; in columns 3,4, two entries (1,2) are included for the number of iterations and cpu-time if FDR is used twice in the example; And we take the max value in the residual error columns 5 and 8; $(s(M), s(\hat{M}))$ is sizes of moment matrix $M$ and facially reduced matrix $\hat{M}$ , resp.; column 7 is the SVD tolerance for GIF and the residual error for the moment matrix using the Interior Point calculation with SeDuMi, DNC - Did Not Converge; the Maple SVD computations in GIF-FDR were executed with tolerance $:= 10^{-10}$ and <i>Digits</i> $:= 15$ , resp. . . . .	82
4.1	<b>Comparison between facial reduction and SeDuMi (1)</b> All data is obtained by using minimal number of facial reductions; Here: min (max) # FR means minimal (maximum) number of facial reductions in our tests; rank(FR) means the size of the problem after each facial reduction, the first one is the size of the original problem; Singlty degree is the singularity degree of the SDP problem after the 1st facial reduction; Res(FR) is the residual of the final moment matrix using facial reduction and DR iterations (Algorithm 4.3.1); Res(CVX) is the residual of the final moment matrix using CVX(SeDuMi). . . . .	119

**4.2 Comparison between facial reduction and SeDuMi (2)** All data obtained here is by using minimal number of facial reductions; max rank is the maximum rank of the moment matrix; res each FR is the residual of solving the corresponding SDP problem by DR after each facial reduction; # DR each FR is the number of DR iterations to solve the corresponding SDP problem after each facial reduction; thres FR is the tolerance to obtain the correct maximum rank using facial reductions (Algorithm 4.3.1); thres CVX is the tolerance to obtain the correct maximum rank using CVX(SeDuMi); . . . 119

# List of Appendices

Appendix . . . . .	133
Appendix . . . . .	135

# Chapter 1

## Introduction

The thesis is aimed at developing theory and numerical algorithms for transforming a system of polynomial equations with real coefficients into an equivalent potent system, called generating polynomial equations for the real radical of the system. Such real radical generating systems enjoy potent properties: they are free of multiplicities which cause ill-conditioning in numerical solution methods; in the case of finitely many solutions they have no non-real solutions; they are free of sums of squares of polynomials. These and other advantages mean that the problem of finding stable and efficient algorithms for the approximation of real radicals is an important open problem, which is the focus of much recent research [16, 20, 25, 6].

Systems of polynomial equations requiring analysis of their real solutions arise in many applications. For example, in mechanism design, the fixed distance between joints are expressed naturally as quadratic equations in the joint coordinates [1]. In chemistry, equilibria of chemical reactions are naturally modelled as solutions of polynomial equations [26]. Biology yields analogous systems and equilibria as real solutions of polynomial equations [22].

Historical high points in polynomial solving are the discovery of formulae for their exact solution in terms of rational functions of the coefficients and radicals for the quadratic, cubic and quartic polynomials. Subsequently Galois and Abel famously showed that such formulae do not exist for univariate polynomials of degree  $\geq 5$ . The mathematical study of such systems and their solutions constitutes *Algebraic Geometry*, one of the foundation areas of Mathematics. Indeed only relatively few polynomials of higher degree are exactly solvable, with Galois giving a criterion for such solvability. Since our focus is on general polynomial systems of

higher degree with approximate real coefficients, the methods developed in the thesis are numerical methods, rather than such exact methods.

This thesis is a multidisciplinary work between the areas of Computer Science, Algebraic Geometry, Numerical Analysis, Convex Optimization, and methods originating in the Geometry of Partial Differential Equations. To help the reader understand the main ideas, in Section 1.1 we will introduce some elementary material on solutions of polynomial systems over the real numbers  $\mathbb{R}^n$  and the complex numbers  $\mathbb{C}^n$ . In Section 1.2, we will introduce material on ideals of polynomial systems over  $\mathbb{R}$  and  $\mathbb{C}$ . The more complicated objects of radicals of these ideals are also introduced in this section, together with simple examples. Section 1.3 will give a simple introduction by examples to characterizing the real radical as the solution of a semi-definite programming (SDP) problem involving a so-called moment matrix. Section 1.4 will give some basic background about SDP problems, their primal and their dual forms, and facial reduction. Section 1.5 gives a higher level view of the moment and moment matrix problem and relevant results in the literature. Section 1.6 gives an outline of the contents of the thesis.

## 1.1 Real and complex solution sets (varieties) of systems of polynomial equations

Throughout this thesis we consider systems of polynomial equations in variables  $x = (x_1, x_2, \dots, x_n)$  which are either real ( $x \in \mathbb{R}^n$ ) or complex ( $x \in \mathbb{C}^n$ ), with coefficients which are usually real or some times complex. Since we are not developing exact methods, we don't consider the case of exact (e.g. integer, rational, or modular) coefficients and focus on the case that mostly occurs in applications, that of real solutions of polynomials with real coefficients.

So we consider systems of polynomials in variables  $x = (x_1, x_2, \dots, x_n)$ :

$$p_1(x) = 0, \dots, p_k(x) = 0 \tag{1.1}$$

where usually the polynomials have real coefficients, that is each polynomial belongs to the polynomial ring  $\mathbb{R}[x]$ , or complex in which case  $P = \{p_1, \dots, p_k\} \subset \mathbb{C}[x]$ . See [8], for definition and discussion of polynomial rings. The solution sets (varieties) over  $\mathbb{C}$  and  $\mathbb{R}$  are naturally defined as follows:

**Definition 1.1.1 (Variety)** Given  $P = \{p_1, \dots, p_k\} \subset \mathbb{R}[x]$  where  $x = (x_1, \dots, x_n)$  we define

$$V_{\mathbb{C}}(P) := \{x \in \mathbb{C}^n : p_1(x) = 0, \dots, p_k(x) = 0\} \quad (1.2)$$

The solution set over  $\mathbb{R}$ , or real variety, is defined as:

$$V_{\mathbb{R}}(P) := \{x \in \mathbb{R}^n : p_1(x) = 0, \dots, p_k(x) = 0\} \quad (1.3)$$

Note that sometimes we will write  $P(x) = 0$  for brevity, or even  $p(x) = 0$  instead of  $p_1(x_1, \dots, x_n) = 0, \dots, p_k(x_1, \dots, x_n) = 0$ . Obviously  $V_{\mathbb{R}}(P) \subseteq V_{\mathbb{C}}(P)$  and the geometry of the varieties can be quite different as the following sum of squares example shows.

**Example 1.1.1** Consider the single equation  $u^2 + v^2 = 0$ . Here  $x_1 = u, x_2 = v$ . Then

$$V_{\mathbb{C}}(u^2 + v^2) := \{(u, v) \in \mathbb{C}^2 : u^2 + v^2 = 0\} = \{(+iv, v) \in \mathbb{C}^2\} \cup \{(-iv, v) \in \mathbb{C}^2\} \quad (1.4)$$

$$V_{\mathbb{R}}(u^2 + v^2) := \{(u, v) \in \mathbb{R}^2 : u^2 + v^2 = 0\} = \{(0, 0)\} \quad (1.5)$$

Here the complex variety is the union of two 1-dimensional manifolds (lines). The real variety is 0-dimensional and consists of a point. To give the reader a brief taste of real radicals, the complex radical for the above example has generator  $u^2 + v^2$  and the real radical has generators  $u, v$  corresponding to the much more pleasant equivalent system of equations  $u = 0, v = 0$ .

The main problem of this thesis, the approximation of the real radical ideal of a system of real polynomials, is motivated by difficulties in the numerical solution of such systems due to multiplicities and sums of squares. So we now give a brief discussion of some numerical methods for solving such systems of equations. One of the oldest methods, Newton's method, is a local method, which provided it is given an initial guess sufficiently close to an isolated solution, and the system is regular enough (e.g. has non-singular Jacobian) will converge to that solution. Variations of such local methods are the most common in applications. The thesis does not focus on such methods, but instead on global methods, which obtain information about the complete set of solutions of a polynomial system.

To discuss recent methods most relevant to the thesis, we first consider polynomial systems in  $\mathbb{C}[x]$  in  $n$  variables  $x = (x_1, x_2, \dots, x_n)$

$$p_1(x) = 0, \dots, p_k(x) = 0 \quad (1.6)$$

with finitely many solutions in  $\mathbb{C}$ . Remarkably, this apparently special subclass forms a building block for the new methods of Complex Numerical Algebraic Geometry, that describe general systems and their solutions. It is shown that all the finitely many solutions can be obtained by continuously deforming the solutions of related (start) system into the target solutions. To give the reader a brief sketch of the main ideas, consider the case, where there is a single polynomial in one variable  $p = 0$  of degree  $d$ . A suitable start system is  $q = \alpha x^d - \beta$  where  $\alpha$  and  $\beta$  are nonzero random constants. The homotopy function can be taken as  $H(x, t) = (1 - t)q + tp$ . Then as the real deformation parameter  $t$  goes from  $t = 0$  to  $t = 1$  it deforms from the exactly solvable start system to the target system. Numerical path tracking methods approximately solve the related differential equation

$$\frac{dH}{dt} = 0 = \frac{\partial H}{\partial x} \frac{dx}{dt} + \frac{\partial H}{\partial t} \quad (1.7)$$

subject to the initial conditions  $x(0)$  being set to the  $d$  exact solutions of the start system. The randomness is needed to ensure that the Jacobian  $\frac{\partial H}{\partial x}$  does not become singular along the path. In the case of the solutions with multiplicities several paths converge to a single solution of the target, and the system (in particular  $\frac{\partial H}{\partial x}$ ) becomes singular at  $t = 0$ . Such singularities caused by multiplicities are one of the motivations for determining the equivalent system of equations constituting the real radical considered in this thesis.

A breakthrough leading to the creation of Complex Numerical Algebraic Geometry, by Sommese and Wampler (see [26, 2] and the references therein), was to reduce the general case with positive dimensional solution manifolds to the above zero dimensional case for square systems. The key idea is to cut the variety with a random linear space, that intersects at so-called witness points. The method involves embedding in square systems, by appending slack variables and extra equations, then slicing with linear spaces to cut out the witness points. A simple example is to consider a single non-constant polynomial  $f(u, v) = 0$  in  $\mathbb{C}[u, v]$ . Then slice it with a random line  $au + bv + c = 0$ . The witness points are solutions of  $f(u, -au/b - c/b) = 0$  which by the Fundamental Theorem of Algebra, has at least one complex root. This root can be approximated by applying the homotopy solver to the 0-dimensional system  $f(u, v) = 0, au + bv + c = 0$  yielding a corresponding witness point on  $V_{\mathbb{C}}(f(u, v))$ . The resulting implemented algorithms, in Bertini and PHCPack [2, 30] have undergone consider-



able development, and theoretically give witness points on every component of the complex variety of a complex system.

However the above method obviously fails when applied to real varieties. For example consider  $f(u, v) = u^2 + v^2 - 1 = 0, au + bv + c = 0$  in  $\mathbb{R}[u, v]$ . Then a random real line can miss the circle with high probability. There have been considerable developments in a method to address this case. Such methods, called Critical Point methods, find critical points of the distance function of a point to a real variety [13] or a hyperplane to a real variety [32]. This yields a 0-dimensional system for the critical points, which can be solved by homotopy continuation. The (real) critical points result from discarding the complex solutions of the 0-dimensional system.

At present although this method in theory gives a critical point on every connected component of a real variety, it can not be called a reliable numerical method, since it may fail due to multiplicities, singularities and sums of squares in the system. For these reasons, it is important to find an equivalent system to the input, that is free of multiplicities, sums of squares, and excess non-real solutions. These are aspects of the generators of the real radical ideal, whose approximate computation is investigated in this thesis. Thus we discuss ideals and their radicals in the next section.

## 1.2 Equivalent systems of polynomials: generators of ideals and radicals of polynomial systems

It is natural and necessary in applications to manipulate systems of polynomial equations into equivalent forms, in which they enjoy better properties (e.g. are easier to solve numerically, lower degree, or aspects of their solutions are more transparent). Such motivations underly polynomial ideal theory.

A polynomial system in  $\mathbb{R}$  or  $\mathbb{C}$  can be viewed as a linear function of its monomials. Therefore it is natural to write it as a matrix equation, and apply linear elimination to the system.

**Example 1.2.1** Consider the system with polynomials  $g_1 = x^8 - 3x^4 + 2$ ,  $g_2 = x^8 - x^4 - 2$ :

$$P = \{g_1, g_2\} \subseteq \mathbb{R}[x] \quad (1.8)$$

Here the coefficient matrix is given by  $C(P)$  below:

$$C(P) \cdot \mathbf{x}^{(\leq 8)} = \begin{pmatrix} -2 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 & -3 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} 1 \\ x^1 \\ \vdots \\ x^7 \\ x^8 \end{bmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (1.9)$$

Gaussian elimination on the coefficient matrix or equivalently in terms of the polynomials yields:  $x^8 - 3x^4 + 2 - (x^8 - x^4 - 2) = -2x^4 + 4$ . So we have obtained an simpler lower degree polynomial  $g_3 = x^4 - 2 = 0$ . To check that  $g_3$  is equivalent to the original system  $\{g_1 = 0, g_2 = 0\}$  we calculate

$$\begin{aligned} g_1 &= x^8 - 3x^4 + 2 = (x^4 - 1)g_3 \\ g_2 &= x^8 - x^4 - 2 = (x^4 + 1)g_3 \end{aligned} \quad (1.10)$$

So the two original polynomials  $g_1, g_2$  are multiples of  $g_3$  and can be discarded. Notice that to discard the original polynomials we need to multiply by monomials of form  $x^\ell$ .

The previous example naturally motivates the definition of a polynomial ideal.

**Definition 1.2.1** A polynomial ideal over a field  $\mathbb{K}$  where  $\mathbb{K} = \mathbb{C}$  or  $\mathbb{R}$  with generators  $\{g_1, g_2, \dots, g_k\} \subset \mathbb{K}[x]$  is the infinite set of polynomials:

$$\langle g_1, \dots, g_k \rangle_{\mathbb{K}} := \{f_1 g_1 + \dots + f_k g_k : f_j \in \mathbb{K}[x], 1 \leq j \leq k\} \quad (1.11)$$

In the above example, by elimination we identified a lower degree generator  $g_3 \in \langle g_1, g_2 \rangle_{\mathbb{R}}$ . Then a further calculation showed that  $g_1 = (x^4 - 1)g_3 \in \langle g_3 \rangle_{\mathbb{R}}$  and  $g_2 = (x^4 + 1)g_3 \in \langle g_3 \rangle_{\mathbb{R}}$ . So we have an equivalent and lower degree generator for the ideal, which has the same real and complex varieties. Sophisticated elimination algorithms have been developed for the multivariate polynomial systems, for reducing the systems to an equivalent set of generators called a Gröbner basis for the ideal. Such bases have the same complex variety as the input system.

These methods rely on Gauss elimination in the exact case, and so are often unstable in the approximate case. Instead, we use Geometric Involutive Bases [12], resulting from concepts in the geometric theory of differential equations, and implemented using stable methods from Numerical Linear Algebra (especially the Singular Value Decomposition). See [8] for modern treatments of Gröbner Bases. For the simple example above  $x^4 - 2$  is both a Gröbner basis and a Geometric Involutive Basis for the  $\langle g_1, g_2 \rangle_{\mathbb{R}}$ .

This thesis is directed towards numerically computing a generating set for a special kind of ideal targeted at real solutions of the input system: *real radical ideals*. There are exact (symbolic) algorithms for finding real radicals, for example, methods developed by Becker & Neuhauser [3] and Spang [28]. However, they are not designed for approximate computation when there are small numerical errors involved in the input.

**Definition 1.2.2 (Real Radical Ideal)** *Given a system of polynomials  $g$  with generators  $g = \{g_1, \dots, g_k\} \subset \mathbb{R}[x]$  the real radical ideal of  $\langle g_1, \dots, g_k \rangle_{\mathbb{R}}$  is defined as*

$$\sqrt[\mathbb{R}]{\langle g_1, \dots, g_k \rangle_{\mathbb{R}}} = \{f(x) \in \mathbb{R}[x] : f(x) = 0 \text{ for all } x \in V_{\mathbb{R}}(g)\} \quad (1.12)$$

A complex radical is defined by replacing  $\mathbb{R}$  in this definition with  $\mathbb{C}$ .

**Example 1.2.2** *Consider a univariate polynomial  $g_1 \in \mathbb{R}[x]$ . To find a generator for  $\sqrt[\mathbb{R}]{\langle g_1 \rangle_{\mathbb{R}}}$  we use the factorization of  $g_1$  over  $\mathbb{R}$ :  $g_1 = \prod_j (x - a_j)^{m_j} \prod_k (x^2 + b_k x + c_k)^{r_k}$  where  $a_j, b_k, c_k$  are all real with  $b_k^2 - 4c_k < 0$  and  $m_j, r_k$  are the respective multiplicities in the factorization. Then*

$$V_{\mathbb{R}}(g_1) = V_{\mathbb{R}}(\prod_j (x - a_j) \prod_k (x^2 + b_k x + c_k)) = V_{\mathbb{R}}(\prod_j (x - a_j)) \quad (1.13)$$

and the real radical of  $g_1$

$$\sqrt[\mathbb{R}]{\langle g_1 \rangle_{\mathbb{R}}} = \langle \prod_j (x - a_j) \rangle_{\mathbb{R}} \quad (1.14)$$

is generated by a polynomial obtained by discarding multiplicities and the factors with non-real roots from  $g_1$ . For the previous example the real variety is given by

$$V_{\mathbb{R}}(x^4 - 2) = V_{\mathbb{R}}((x^2 - \sqrt{2})(x^2 + \sqrt{2})) = V_{\mathbb{R}}(x^2 - \sqrt{2}) \quad (1.15)$$

and so its real radical is generated by  $x^2 - \sqrt{2}$  which has no multiplicities and only real roots. Thus  $\sqrt[\mathbb{R}]{\langle x^4 - 2 \rangle_{\mathbb{R}}} = \langle x^2 - \sqrt{2} \rangle_{\mathbb{R}}$ . There are various equivalent forms of the real radical, and

*complex radical. We have only given one, to communicate the main ideas in a simplified way in this introduction. In the later chapters some of these equivalent forms will be given, when they are needed in proofs and other material.*

Another motivation for computing the generators of the real radical ideals is to verify the completeness of a real solution set of a given polynomial system. Given a polynomial system  $g$ , suppose  $s \subseteq V_{\mathbb{R}}(g)$ . The completeness of  $s$  means the *Zariski* closure,  $\bar{s}$ , is equal to the real variety  $V_{\mathbb{R}}(g)$ . First we have  $\bar{s} \subseteq V_{\mathbb{R}}(g)$ . By computing the generators of the real radical ideal, we can verify  $I(s) \subseteq \sqrt[\mathbb{R}]{\langle g \rangle_{\mathbb{R}}}$  which indicates  $V_{\mathbb{R}}(g) \subseteq \bar{s}$ , thus we know  $\bar{s} = V_{\mathbb{R}}(g)$ .

There are symbolic methods [23] for the computation of the generators of real radical ideal. One can also use methods involving *triangular decomposition of semi-algebraic sets* to compute the connected components of the real variety. However, these methods are exact methods and they are not stable for numerical computations with approximate coefficients. For a comparison with triangular decomposition of semi-algebraic sets, see chapter 4.

A fundamental open problem is to generalize the work of [16, 27] to positive dimensional ideals. The algorithm of [19, 20] for a given input real polynomial system  $P$ , modulo the successful application of SDP methods at each of its steps, computes a Pommaret basis  $Q$ :

$$\sqrt[\mathbb{R}]{\langle P \rangle_{\mathbb{R}}} \supseteq \langle Q \rangle_{\mathbb{R}} \supseteq \langle P \rangle_{\mathbb{R}} \quad (1.16)$$

and would provide a solution to this open problem if it is proved that  $\langle Q \rangle_{\mathbb{R}} = \sqrt[\mathbb{R}]{\langle P \rangle_{\mathbb{R}}}$ . We believe that the work [19, 20] establishes an important feature – involutivity – that will necessarily be a main condition of any theorem and algorithm characterizing the real radical. Involutivity is a natural condition, since any solution of the above open problem using SDP, if it establishes radical ideal membership, will necessarily need (at least implicitly) a real radical Gröbner basis. Our algorithm, uses geometric involutivity, and similarly gives an intermediate ideal, which constitutes another variation on this family of conjectures.

### 1.3 Introductory example of computation of the real radical using Moment Matrices and SDP

We give an example in this section so the readers can have a preliminary outline of how to use the moment matrix to compute the real radical ideal. For a theoretical introduction, see Section 1.5.

Suppose a degree 4 polynomial  $p = x^4 - 2$  is given and we wish to reproduce the result we found from the complete factorization in the previous section. In matrix form, the polynomial is represented by its coefficient matrix  $B = [-2, 0, 0, 0, 1]^T$ .

The truncated moment matrix is a  $5 \times 5$  matrix whose  $(\alpha, \beta)$  entry is  $u_{\alpha+\beta}$  corresponding to  $x^\alpha x^\beta$  and  $\alpha, \beta \in \mathbb{N}_4$  given by:

$$M = \begin{pmatrix} u_0 & u_1 & u_2 & u_3 & u_4 \\ u_1 & u_2 & u_3 & u_4 & u_5 \\ u_2 & u_3 & u_4 & u_5 & u_6 \\ u_3 & u_4 & u_5 & u_6 & u_7 \\ u_4 & u_5 & u_6 & u_7 & u_8 \end{pmatrix} \quad (1.17)$$

In the SDP-moment matrix approach the given polynomial system, in this case  $\{x^4 - 2\}$ , is first prolonged to degree 8 by multiplying  $x, x^2, x^3, x^4$ :

$$\{x^4 - 2, x^5 - 2x, x^6 - 2x^2, x^7 - 2x^3, x^8 - 2x^4\}. \quad (1.18)$$

The constraint system imposed on the moment matrix, assuming  $u_0 = 1$ , is equivalent to  $B^T \cdot M = 0$  or the following linear system

$$u_4 - 2 = 0, u_5 - 2u_1 = 0, u_6 - 2u_2 = 0, u_7 - 2u_3 = 0, u_8 - 2u_4 = 0 \quad (1.19)$$

Imposing these constraints the truncated moment matrix  $M$  is

$$M = \begin{pmatrix} 1 & u_1 & u_2 & u_3 & 2 \\ u_1 & u_2 & u_3 & 2 & 2u_1 \\ u_2 & u_3 & 2 & 2u_1 & 2u_2 \\ u_3 & 2 & 2u_1 & 2u_2 & 2u_3 \\ 2 & 2u_1 & 2u_2 & 2u_3 & 4 \end{pmatrix} \quad (1.20)$$

We then solve an SDP optimization problem to compute a generic point  $(u_1, u_2, u_3)$  if possible such that  $M$  is a positive semidefinite matrix with maximum rank. A solution is  $(u_1, u_2, u_3) = (0, \sqrt{2}, 0)$ . Its associated moment matrix and moment matrix kernel are:

$$M = \begin{pmatrix} 1 & 0 & \sqrt{2} & 0 & 2 \\ 0 & \sqrt{2} & 0 & 2 & 0 \\ \sqrt{2} & 0 & 2 & 0 & 2\sqrt{2} \\ 0 & 2 & 0 & 2\sqrt{2} & 0 \\ 2 & 0 & 2\sqrt{2} & 0 & 4 \end{pmatrix}, \ker M = \text{span}_{\mathbb{R}} \left\{ \begin{pmatrix} -2 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -\sqrt{2} \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -\sqrt{2} \\ 0 \\ 1 \\ 0 \end{pmatrix} \right\} \quad (1.21)$$

The kernel corresponds to the generating set

$$\{\sqrt{2} - x^2, 2 - x^4, \sqrt{2}x - x^3\}. \quad (1.22)$$

The last two polynomials are consequences of  $\sqrt{2} - x^2$  multiplying by  $\sqrt{2} + x^2$  and  $x$ , so are discarded, since they lie in  $\langle \sqrt{2} - x^2 \rangle_{\mathbb{R}}$ . By Laurent and Rostalski [18]  $\sqrt{2} - x^2$  is indeed a basis of the real radical of  $2 - x^4$ , as we found from the complete factorization in Section 1.2.

## 1.4 SDP optimization

In this section, we discuss semidefinite matrices and semidefinite programs (SDP). We introduce the semidefinite duality theory and facial structure theory of SDP cones [31].

### 1.4.1 Semidefinite Matrices

A symmetric matrix  $M$  of size  $n \times n$  is called *positive semidefinite*, denoted as  $M \geq 0$ , if one of the following two equivalent criteria is satisfied:

1.  $x^T M x \geq 0$  for all  $x \in \mathbb{R}^n$ .
2. All eigenvalues of  $M$  are non-negative.

Similarly, a symmetric matrix  $M$  of size  $n \times n$  is called *positive definite*, denoted as  $M > 0$ , if one of the following two equivalent criteria is satisfied:

1.  $x^T M x > 0$  for all non-zero  $x \in \mathbb{R}^n$ .
2. All eigenvalues of  $M$  are strictly positive.

The set of all  $n \times n$  symmetric matrices is denoted as  $\mathcal{S}^n$ . The cone of all  $n \times n$  positive semidefinite matrices is denoted as  $\mathcal{S}_+^n$ . The cone of all  $n \times n$  positive definite matrices is denoted as  $\mathcal{S}_{++}^n$ .

**Definition 1.4.1 (Trace product)** *Given two symmetric matrices  $A, B$ , we define the trace inner product  $\langle A, B \rangle = \text{trace}(A^T B) = \sum_{ij} A_{ij} B_{ij}$ .*

## 1.4.2 Semidefinite Programs

There are two forms of writing semidefinite programs. Given  $A_1, A_2, \dots, A_m, C, X \in \mathcal{S}^n$  and  $b_1, b_2, \dots, b_m \in \mathbb{R}$ . Define the linear operator:  $\mathcal{A}(X) = [\langle A_1, X \rangle, \langle A_2, X \rangle, \dots, \langle A_m, X \rangle]^T$ . Let  $b = [b_1, b_2, \dots, b_m]^T$ .

The primal form of an SDP is written as:

$$\begin{aligned} & \min \langle C, X \rangle \\ \text{s.t. } & \mathcal{A}(X) = b \\ & X \geq 0. \end{aligned} \tag{1.23}$$

The dual form of an SDP is written as:

$$\begin{aligned} & \max b^T y \\ \text{s.t. } & Z = C - \sum_{i=1}^m A_i y_i \\ & Z \geq 0 \end{aligned} \tag{1.24}$$

The adjoint of  $\mathcal{A}$  is defined to be  $\mathcal{A}^* y = \sum_{i=1}^m A_i y_i$ .

**Theorem 1.4.1 (Weak Duality [31])** *If  $X$  is feasible for the primal SDP and  $(y, Z)$  are feasible for the dual SDP, then  $\langle C, X \rangle \geq b^T y$ .*

### 1.4.3 Face, minimal face and facial structure

We give a brief introduction to faces, minimal faces, and lemmas about facial structure. The definitions below can be found in [4, 5, 7, 11, 24].

**Definition 1.4.2** *Given convex cones  $F, K$  and  $F \subseteq K$ , we call  $F$  a face of  $K$ , and write  $F \trianglelefteq K$ , if*

$$x, y \in K, x + y \in F \implies x, y \in F.$$

*Given a nonempty convex subset  $S$  of  $K$ , the minimal face of  $K$  containing  $S$  is defined to be the intersection of all faces of  $K$  containing  $S$ .*

**Definition 1.4.3** *Suppose  $F$  is a face of  $\mathcal{S}_+^n$ . The orthogonal complement of  $F$ , denoted as  $F^\perp$ , is defined to be  $F^\perp = \{Z \in \mathcal{S}^n : Z \cdot X = 0, \forall X \in F\}$ . The dual cone of  $F$ , denoted as  $F^*$ , is defined to be  $F^* = \{Z \in \mathcal{S}^n : Z \cdot X \geq 0, \forall X \in F\}$ .*

The following lemmas about the facial structure of the semidefinite cone  $\mathcal{S}_+^n$  are well-known, see e.g. [31].

**Lemma 1.4.2** *Any face  $F$  of  $\mathcal{S}_+^n$  is either  $0$ ,  $\mathcal{S}_+^n$  or*

$$F = \{X \in \mathcal{S}^n : X = U M U^T, M \in \mathcal{S}_+^r\} \quad (1.25)$$

*where  $U$  is an  $n \times r$  matrix and  $U^T U = I$ .*

**Lemma 1.4.3** *Suppose  $F$  is a face of  $\mathcal{S}_+^n$  and  $W \in \mathcal{S}_+^n$ . Then  $F \cap \{W\}^\perp$  are faces of  $\mathcal{S}_+^n$ , where  $\{W\}^\perp = \{X \in \mathcal{S}^n : X \cdot W = 0\}$ .*

### 1.4.4 Facial reduction

The idea of facial reduction was originally developed by Borwein and Wolkowicz [4, 5] in the 1980s. However it has been nontrivial to develop practical algorithms implementing facial reduction. Only recently have practical algorithms been developed. For example it was recently applied to solve the large sensor network localization problems [15, 10].

We consider the set  $F_P = \{X \in \mathcal{S}^n : \mathcal{A}(X) = b, X \geq 0\}$  which has the same form as the feasible set of the moment matrix SDP optimization problem considered in this thesis, clearly



$F_P$  is a convex subset of  $\mathcal{S}^n$ . The following theorem gives information on the facial structure of  $F_P$ :

**Lemma 1.4.4 (Facial reduction [24])** *Define  $F_{\min}$  to be the minimal face containing  $F_P$ . Let  $\mathcal{A}^*$  be the adjoint of  $\mathcal{A}$  defined before. For a face  $F \trianglelefteq \mathcal{S}_+^n$  containing  $F_P$ , the following holds :*

$$\left\{ \begin{array}{l} (I) \quad \mathcal{A}(X) = b, X \in F \\ (II) \quad b^T y = 0, Z = \mathcal{A}^* y \in F^* \setminus F^\perp \end{array} \right\} \Rightarrow X \in \{Z\}^\perp \cap F \subset F. \quad (1.26)$$

*In addition,  $F = F_{\min}$  if and only if (II) has no solution.*

The matrix  $Z$  is called the *exposing vector* of  $F$ . Each time (II) is solved, an exposing vector  $Z$  is obtained and can be used to update  $F \leftarrow \{Z\}^\perp \cap F$ . Repeating this process until (II) is infeasible ((II) admits no solution), we get a sequence of faces containing  $F_P$ :  $F_0 \supset F_1 \supset F_2 \supset \dots \supset F_{\min} \supset F_P$  where  $F_0 = \mathcal{S}_+^n$  and  $F_{i+1} = F_i \cap \{Z_i\}^\perp$ . This iteration process to find the minimal face  $F_{\min}$  is called *facial reduction* on the primal form and is guaranteed to terminate in at most  $n - 1$  iterations [29]. The minimal number of facial reductions is called the *singularity degree*.

## 1.5 Moment problem

In this section, we briefly introduce some background and results about the classical moment problem and moment matrices. We also discuss how semidefinite moment matrices are connected to real radical ideals. Most of the results are from Curto & Fialkow [9] and Lasserre, Laurent & Rostalski [17], [18]. For the proofs of the theorems, please see the corresponding references. For background knowledge about semidefinite programming, see Section 1.4.

### 1.5.1 Linear form, positive linear form and moment matrix

**Definition 1.5.1** *Given a linear form  $\lambda \in \mathbb{R}[x]^*$ ,  $\lambda$  is said to be positive written  $\lambda \geq 0$  if  $\lambda(f^2) \geq 0$  for all  $f \in \mathbb{R}[x]$ . Here  $x = (x_1, \dots, x_n)$  and  $\mathbb{R}[x]^*$  is the dual space representing functionals from  $\mathbb{R}[x]$  to  $\mathbb{R}$ .*

**Definition 1.5.2** *Define the quadratic form  $Q_\lambda$  such that  $Q_\lambda(f) = \lambda(f^2)$ . Define the kernel of  $Q_\lambda$  to be  $\ker Q_\lambda = \{f \in \mathbb{R}[x] : Q_\lambda(f) = 0\}$ .  $Q_\lambda$  is said to be positive semidefinite if  $Q_\lambda(f) \geq 0$ .*

The quadratic form  $Q_\lambda$  can be extended to a bilinear form such that  $Q_\lambda(f, g) = \lambda(fg)$ .

**Definition 1.5.1 (Moment Matrix [18])** *Given a linear form  $\lambda \in \mathbb{R}[x]^*$ ,  $x = (x_1 \cdots x_n)$  which maps a polynomial to a real number. A symmetric infinite matrix*

$$M(\lambda) = (\lambda(x^\alpha x^\beta))_{\alpha, \beta \in \mathbb{N}^n} \quad (1.27)$$

*is called a moment matrix of  $\lambda$  where  $\mathbb{N} = \{0, 1, 2, \dots\}$ . We use graded lexicographic order for  $\alpha$  and  $\beta$  throughout this thesis.*

**Example 1.5.1** *Consider  $\lambda = \mathbb{R}[x, y]^*$  such that  $\lambda = \frac{1}{2}\lambda_{1,2} + \frac{1}{2}\lambda_{2,1}$  ( $\lambda_{1,2}$  is the evaluation at  $x = 1, y = 2$  and  $\lambda_{2,1}$  is the evaluation at  $x = 2, y = 1$ ). Let  $v = [1, x, y, x^2, xy, y^2, \dots]^T$*

$$M(\lambda) = \begin{bmatrix} 1 & \frac{3}{2} & \frac{3}{2} & \cdots \\ \frac{3}{2} & \frac{5}{2} & 2 & \cdots \\ \frac{3}{2} & 2 & \frac{5}{2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} = \lambda(v \cdot v^T). \quad (1.28)$$

**Theorem 1.5.1** [18] *Given a moment matrix  $M(\lambda)$  corresponding to  $\lambda$ , We have  $\lambda(f^2) = \text{vec}(f) \cdot M(\lambda) \cdot \text{vec}(f)^T$ . In addition,  $M(\lambda) \geq 0$  if and only if  $\lambda$  is positive ( $Q_\lambda$  is positive semidefinite).*

**Theorem 1.5.2** *Suppose  $\lambda \geq 0$ . Then a polynomial  $p$  belongs to  $\ker Q_\lambda$  if and only if its coefficient vector belongs to  $\ker M(\lambda)$ . That is, we have  $\ker Q_\lambda = \ker M(\lambda)$ .*

**Proof** Given  $\lambda \geq 0$ , we have  $M(\lambda) \geq 0$ . So  $Q_\lambda(f) = \lambda(f^2) = 0$  implies  $\text{vec}(f) \cdot M(\lambda) \cdot \text{vec}(f)^T = 0$ . Since  $M(\lambda) \geq 0$ ,  $M(\lambda)$  has a Cholesky factorization  $M(\lambda) = BB^T$ . So  $\text{vec}(f)B(\text{vec}(f)B)^T = 0$  which means  $\text{vec}(f)B = 0$  and  $M(\lambda) \cdot \text{vec}(f)^T = 0$ .

**Theorem 1.5.3** *Suppose  $\lambda \geq 0$ . Then  $\ker Q_\lambda$  is an ideal, which is also real radical.*

**Proof** Suppose  $f \in \ker Q_\lambda$  and  $g$  is an arbitrary polynomial, we need to show that  $fg \in \ker Q_\lambda$  as well. Now  $\lambda(f^2) = 0$  implies  $\text{vec}(f) \cdot M(\lambda) \cdot \text{vec}(f)^T = 0$  and  $\lambda \geq 0$  implies  $M(\lambda) \geq 0$ . So we have  $M(\lambda) \cdot \text{vec}(f)^T = 0$ . From the structure of the moment matrix, it means  $\lambda(x^\alpha f) = 0$  for any monomial  $x^\alpha \in \mathbb{R}[x]$ . So  $\lambda(f^2 g^2) = \lambda(f g^2 \cdot f) = \lambda((m_1 + \cdots + m_n)f) = 0$  where  $m_1, \dots, m_n$  are monomials of  $f g^2$ . So  $fg \in \ker Q_\lambda$ .

The proof of the real radical property can be found in [18].

### 1.5.2 Moment Problem

**Theorem 1.5.4 (Riesz-Haviland's Theorem [14])** *For a linear form  $\lambda \in \mathbb{R}[x]^*$  and closed set  $K$  in  $\mathbb{R}^n$ , the following two conditions are equivalent:*

- $\lambda(f) \geq 0$  for all  $f \in \mathbb{R}[x]$  such that  $f \geq 0$  on  $K$
- There is a (positive) Borel measure  $\mu$  on  $K$  such that  $\lambda(f) = \int_K f d\mu$  for all  $f \in \mathbb{R}[X]$ .

However, a nonnegative polynomial over  $\mathbb{R}^n$  need not to be a sum of squares of polynomials except in the univariate case (Hilbert 17th problem). For example the *Motzkin polynomial* given by T.Motzkin [21]. It is the polynomial  $F(x, y) = x^4y^2 + x^2y^4 + 1 - 3x^2y^2$ . (The non-negativity of  $F(x, y)$  comes from the arithmetic-geometric mean inequality. Assume  $F(x, y) = \sum_j f_j^2$  is a sum of squares of real polynomials. Then  $\sum_j f_j^2(x, 0) = M(x, 0) = 1$ , which means  $f_j(x, 0)$  are constants. Similarly  $f_j(0, y)$  are constants. Hence each  $f_j$  is of the form  $f_j = a_j + b_jxy + c_jx^2y + d_jxy^2$ . By equating the coefficients of  $x^2y^2$ , we have  $\sum_j b_j^2 = -3$  which is impossible.) So a natural question to ask is when does positivity of  $\lambda$  on sums of squares indicate an integral representation with Borel measure?

Curto and Fialkow show the equivalence in the case that when  $M(\lambda)$  has finite rank, or  $\dim(R[x]/\ker Q_\lambda) = \text{rank}(M(\lambda))$  is finite.

**Theorem 1.5.5 (Curto and Fialkow [9])** *Assume that  $\lambda \geq 0$  and  $\text{rank}(M_\lambda) = r < +\infty$ . Then  $\lambda = \sum_{i=1}^r \alpha_i \lambda_{v_i}$  for some distinct  $v_1, \dots, v_r \in \mathbb{R}^n$  and some real numbers  $\alpha_i > 0$ .  $\lambda_{v_i}$  are evaluations such that  $\lambda_{v_i}(f) = f(v_i)$ . Moreover,  $\{v_1, \dots, v_r\} = V_{\mathbb{R}}(\ker M(\lambda))$ .*

### 1.5.3 Truncated Moment matrix and flat extension theorem

Suppose  $\mathbb{R}[x]_{2d} = \{f \in \mathbb{R}[x] \mid \deg(f) \leq 2d\}$ , we can define the truncated linear form  $\lambda_d \in \mathbb{R}[X]_{2d}^*$  such that  $\lambda_d = \lambda|_{\mathbb{R}[X]_{2d}}$ , the associated quadratic form  $Q_{\lambda_d}$  and the truncated moment matrix  $M(\lambda_d)$ . Similarly, we define the truncated moment matrix.

**Definition 1.5.2 (Truncated Moment Matrix [18])** *Given a linear form  $\lambda_d \in (\mathbb{R}[x]_{2d})^*$ , the truncated moment matrix of  $\lambda_d$  is defined to be*

$$M(\lambda_d) = (\lambda_d(x^\alpha x^\beta))_{\alpha, \beta \in \mathbb{N}_d^n} \quad (1.29)$$

where  $\mathbb{N}_d^n = \{\gamma \in \mathbb{N}^n : |\gamma| = \sum_{j=1}^n \gamma_j \leq d\}$ .

Similarly, we have the following theorems for truncated linear forms and truncated moment matrices.

**Theorem 1.5.6** [18] *Given a truncated moment matrix  $M(\lambda_d)$  corresponding to  $\lambda_d \in \mathbb{R}[x]_{2d}^*$ ,  $M(\lambda_d) \geq 0$  (positive semidefinite) if and only if  $\lambda_d \in \mathbb{R}[x]_{2d}^*$  is positive.*

**Theorem 1.5.7** [18] *A polynomial  $p \in \mathbb{R}[x]_d$  belongs to  $\ker Q_{\lambda_d}$  if and only if its coefficient vector belongs to  $\ker M(\lambda_d) \in \mathbb{R}[x]_d$ .*

**Example 1.5.2** *Suppose  $\lambda_1 \in \mathbb{R}[x, y]_2^*$  and  $\lambda_1(x^a y^b) = u_{a,b}$ . Then*

$$M(\lambda_1) = \begin{bmatrix} u_{00} & u_{10} & u_{01} \\ u_{10} & u_{20} & u_{11} \\ u_{01} & u_{11} & u_{02} \end{bmatrix} \quad (1.30)$$

Without loss, we assume  $u_{00} = 1$ .

The kernel of a positive semidefinite truncated moment matrix has the following “real radical-like” property:

**Lemma 1.5.8** [18] *Assume  $M(\lambda_d) \geq 0$  and let  $p, q_j \in \mathbb{R}[x]$ ,  $f := p^{2m} + \sum_j q_j^2$  with  $m \in \mathbb{N}$ ,  $m \geq 1$ . Then,  $f \in \ker M(\lambda_d) \Rightarrow p \in \ker M(\lambda_d)$ .*

It also has the following ‘ideal-like’ property:

**Lemma 1.5.9 (Moment structure theorem, [18])** *Let  $\lambda_d \in \mathbb{R}[x]_{2d}^*$  and  $f, g \in \mathbb{R}[x]$ ,  $f \in \ker M(\lambda_d)$ .*

(i) *Assume  $M(\lambda_d) \geq 0$ . Then  $\ker M(\lambda_{d-1}) \subseteq \ker M(\lambda_d)$  and  $fg \in \ker M(\lambda_d)$  if  $\deg(fg) \leq d - 1$ .*

(ii) *Assume  $\text{rank } M(\lambda_d) = \text{rank } M(\lambda_{d-1})$ . Then  $\ker M(\lambda_{d-1}) \subseteq \ker M(\lambda_d)$  and  $fg \in \ker M(\lambda_d)$  if  $\deg(fg) \leq d$ .*

The ideal-like property is denoted as the RG condition in the works of Curto and Fialkow [9].

**Definition 1.5.3 (ideal-like condition (RG condition))**

$$f, g \in \mathbb{R}[x], \deg(fg) \leq d, f \in \ker M(\lambda_d) \Rightarrow fg \in \ker M(\lambda_d)$$

**Theorem 1.5.10 (Flat extension theorem, [9])** *Assume  $M(\lambda_d) \geq 0$ . The following statements are equivalent:*

- (i) *There exists an extension of  $M(\lambda_d)$  onto  $M(\lambda_{d+1})$  such that  $M(\lambda_{d+1}) \geq 0$  and  $\text{rank } M(\lambda_d) = \text{rank } M(\lambda_{d+1})$*
- (ii)  *$\ker M(\lambda_d)$  satisfies condition RG.*

**Lemma 1.5.11** [18] *Assume  $M(\lambda) \geq 0$  and  $\text{rank } M(\lambda_d) = \text{rank } M(\lambda_{d-1}) = r$ . Then  $J = \langle \ker M(\lambda_d) \rangle$  is real radical and zero-dimensional. One can extend  $\lambda_d$  to  $\bar{\lambda}$  such that  $\bar{\lambda} \in \mathbb{R}[x]^*$ . Then  $\lambda$  is of the form  $\lambda = \sum_{i=1}^r \alpha_i \lambda_{v_i}$ , where  $\alpha_i > 0$  and  $\{v_1, \dots, v_r\} = V_R(\ker M(\lambda_d))$ .  $\lambda_{v_i}$  are evaluations such that  $\lambda_{v_i}(f) = f(v_i)$ .  $\lambda = \lambda_d$  when restricted to  $\mathbb{R}[x]_{2d}$ .*

### 1.5.4 Generic linear forms

Assume an ideal  $I = \langle h_1, \dots, h_m \rangle_{\mathbb{R}}$ . For  $d \in \mathbb{N}$ , define the set

$$\mathcal{H}_d(I) = \{h_i x^\alpha \mid i = 1, \dots, m, |\alpha| \leq 2d - \deg(h_i)\} \quad (1.31)$$

Define the set

$$\mathcal{K}_d(I) = \{\lambda_d \in \mathbb{R}[x]_{2d}^* \mid \lambda_d(1) = 1, M(\lambda_d) \geq 0 \text{ and } \lambda_d(f) = 0 \forall f \in \mathcal{H}_d(I)\} \quad (1.32)$$

**Theorem 1.5.12** [18] *Suppose  $N_d(I) = \langle \ker M(\lambda_d) \rangle$  and  $\lambda_d$  is a generic linear form (maximum rank) in  $\mathcal{K}_d(I)$ . Then  $N_d(I)$  is independent of the particular choice of the generic element  $\lambda_d \in \mathcal{K}_d(I)$ .*

**Theorem 1.5.13** [18] *We have:  $N_d(I) \subseteq N_{d+1}(I) \subseteq \dots \subseteq \sqrt[d]{I}$ , with equality  $\langle N_d(I) \rangle_{\mathbb{R}} = \sqrt[d]{I}$  for  $d$  large enough.*

## 1.6 Outline of the contents of the thesis

This section gives an outline of the contents of the thesis.

### 1.6.1 Contents of Chapter 2

Geometric involutive bases for polynomial systems of equations have their origin in the prolongation and projection methods of the geometers Cartan and Kuranishi for systems of PDE. They are useful for numerical ideal membership testing and the solution of polynomial systems. In this chapter we further develop our symbolic-numeric methods for such bases. We give methods to explicitly extract and decrease the degree of intermediate systems and the output basis. Algorithms for the numerical computation of involutivity criteria for positive dimensional ideals are also discussed.

We were also motivated by some remarkable recent work by Lasserre and collaborators who employed our prolongation projection involutive criteria as a part of their semi-definite based programming (SDP) method for identifying the real radical of zero dimensional polynomial ideals. Consequently in this chapter we begin an exploration of the interaction between geometric involutive bases and these methods particularly in the positive dimensional case. Motivated by the extension of these methods to the positive dimensional case we explore the interplay between geometric involutive bases and the new SDP methods.

### 1.6.2 Contents of Chapter 3

For a real polynomial system with finitely many complex roots, the real radical ideal, RRI, is generated by a lower degree system that has only real roots and the roots are free of multiplicities. The RRI is a central object in computational real algebraic geometry. The computation of such RRI is of practical interest since multiplicities of roots yield singular Jacobians and cause problems for numerical solvers. Moreover the number of real roots can be far less than the number of complex roots and Lasserre and co-authors have shown that the RRI of a 0-dimensional real polynomial system with finitely many real solutions can be determined by a combination of techniques from a semidefinite programming (SDP) feasibility problem and geometric involution. A conjectured extension of such methods to positive dimensional polynomial systems has been given recently by Ma, Wang and Zhi.

In this section we show that regularity in the form of the Slater constraint qualification (strict feasibility) fails for the moment matrix in the SDP feasibility problem. We use facial

reduction and obtain a smaller regularized problem for which strict feasibility holds. We use this framework for analyzing RRI's of 0 and positive dimensional real polynomial systems. The SDP methods are implemented in MATLAB and our geometric involutive form is implemented in Maple. We consider two approaches to find a feasible moment matrix. We compare the SeDuMi interior point approach within the YALMIP package for MATLAB with the Douglas-Rachford (DR) projection-reflection method.

Illustrative examples show the advantages of the DR approach for some problems over standard interior point methods. We also see the advantage of facial reduction both in regularizing the problem and also in reducing the dimension of the moment matrices.

### 1.6.3 Contents of Chapter 4

Recent breakthroughs have been made in the use of semidefinite programming and its application to real polynomial solving. For example, the real radical of a zero dimensional ideal, can be determined by such approaches as shown by Lasserre and collaborators. Some progress has been made on the determination of the real radical in positive dimension by Ma, Wang and Zhi. Such work involves the determination of maximal rank semidefinite moment matrices. Existing methods are computationally expensive and have poorer accuracy on larger examples.

In previous work we showed that regularity in the form of the Slater constraint qualification (strict feasibility) fails for the moment matrix in the SDP feasibility problem. We used facial reduction to obtain a smaller regularized problem for which strict feasibility holds. However we did not give a theoretical guarantee that our methods, based on facial reduction and Douglas-Rachford iteration ensured the satisfaction of the maximum rank condition to possibly approximate the real radical characterizing all real roots.

This chapter is motivated by the problems above. We discuss how to compute the moment matrix and its kernel using facial reduction techniques where the maximum rank property can be guaranteed by solving the dual problem. The facial reduction algorithms on the primal form is presented. We give examples that exhibit for the first time additional facial reductions beyond the first which can be computed in practice.

Based on these methods and results of Lasserre and collaborators, and Curto and Fialkow,

we give and prove an algorithm for computing the real radical up to any given finite degree. We also prove results regarding the well-posedness of our approach.

## 1.6.4 Conclusions are given in Chapter 5

## 1.6.5 Appendices

## Bibliography

- [1] Teijo Arponen, Samuli Piipponen, and Jukka Tuomela. Kinematic analysis of bricards mechanism. *Nonlinear Dynamics*, 56(1-2):85–99, 2009. 1
- [2] Daniel J Bates, Jonathan D Hauenstein, Andrew J Sommese, and Charles W Wampler. *Numerically solving polynomial systems with Bertini*, volume 25. SIAM, 2013. 4, 25
- [3] Eberhard Becker and Rolf Neuhaus. Computation of real radicals of polynomial ideals. In *Computational algebraic geometry*, pages 1–20. Springer, 1993. 7
- [4] J.M. Borwein and H. Wolkowicz. Facial reduction for a cone-convex programming problem. *J. Austral. Math. Soc. Ser. A*, 30(3):369–380, 1980/81. 12, 54, 94, 98, 106
- [5] J.M. Borwein and H. Wolkowicz. Regularizing the abstract convex program. *J. Math. Anal. Appl.*, 83(2):495–530, 1981. 12, 54, 94, 98, 106
- [6] D. Brake, J. Hauenstein, and A. Liddell. Numerically validating the completeness of the real solution set of a system of polynomial equations. *Proceedings of the 41th International Symposium on Symbolic and Algebraic Computation*, 2016. 1, 115, 118, 121, 130
- [7] Y-L. Cheung, S. Schurr, and H. Wolkowicz. Preprocessing and regularization for degenerate semidefinite programs. In D.H. Bailey, H.H. Bauschke, P. Borwein, F. Garvan, M. Thera, J. Vanderwerff, and H. Wolkowicz, editors, *Computational and Analytical Mathematics, In Honor of Jonathan Borwein's 60th Birthday*, volume 50 of *Springer Proceedings in Mathematics & Statistics*, pages 225–276. Springer, 2013. 12, 70, 72, 97, 98



- [8] David Cox, John Little, and Donal O'shea. *Ideals, varieties, and algorithms*, volume 3. Springer, 1992. 2, 7, 113
- [9] RE Curto and LA Fialkow. Solution of the truncated complex moment problem for flat data-introduction. *Memoirs of the American Mathematical Society*, 119(568):1, 1996. 13, 15, 16, 17, 25, 108, 109, 110
- [10] D. Drusvyatskiy, N. Krislock, Y-L. Cheung Voronin, and H. Wolkowicz. Noisy sensor network localization: robust facial reduction and the Pareto frontier. Technical report, University of Waterloo, Waterloo, Ontario, 2014. arXiv:1410.6852, 20 pages. 12, 54, 94
- [11] D. Drusvyatskiy, G. Pataki, and H. Wolkowicz. Coordinate shadows of semi-definite and euclidean distance matrices. *Math. Programming*, 25(2):1160–1178, 2015. ArXiv:1405.2037.v1. 12, 98
- [12] V.P. Gerdt and Y.A. Blinkov. Involutive bases of polynomial ideals. *Mathematics and Computers in Simulation*, 45(5):519–541, 1998. 7, 24, 29, 59, 112
- [13] Jonathan D Hauenstein. Numerically computing real points on algebraic sets. *Acta applicandae mathematicae*, 125(1):105–119, 2013. 5, 26, 83, 85, 121
- [14] E. K. Haviland. On the momentum problem for distribution functions in more than one dimension. ii. *American Journal of Mathematics*, 58(1):164–168, 1936. 15
- [15] N. Krislock and H. Wolkowicz. Explicit sensor network localization using semidefinite representations and facial reductions. *SIAM Journal on Optimization*, 20(5):2679–2708, 2010. 12, 54, 94
- [16] J.B. Lasserre, M. Laurent, and P. Rostalski. A prolongation–projection algorithm for computing the finite real variety of an ideal. *Theoretical Computer Science*, 410(27):2685–2700, 2009. 1, 8, 25, 47, 53, 54, 55, 64, 84, 93, 94, 120
- [17] Jean Bernard Lasserre, Monique Laurent, and Philipp Rostalski. Semidefinite characterization and computation of zero-dimensional real radical ideals. *Foundations of Computational Mathematics*, 8(5):607–647, 2008. 13, 108, 110, 128

- [18] M. Laurent and P. Rostalski. The approach of moments for polynomial equations. In Miguel F. Anjos and Jean B. Lasserre, editors, *Handbook on semidefinite, conic and polynomial optimization*, International Series in Operations Research & Management Science, 166, pages 25–60. Springer, New York, 2012. 10, 13, 14, 15, 16, 17, 36, 38, 40, 46, 47, 58, 95, 96, 128, 129
- [19] Y. Ma. *Polynomial Optimization via Low-rank Matrix Completion and Semidefinite Programming*. PhD thesis, Academy of Mathematics and Systems Science, Chinese Academy of Science, 2012. 8, 53, 55, 62, 77, 84, 93, 120, 121
- [20] Y. Ma, C. Wang, and L. Zhi. A certificate for semidefinite relaxations in computing positive dimensional real varieties. *Journal of Symbolic Computation*, 72:1 – 20, 2016. vii, x, 1, 8, 53, 55, 62, 77, 78, 79, 80, 81, 82, 84, 93, 120, 121, 128, 129
- [21] Theodore Samuel Motzkin. The arithmetic-geometric inequality. In: *Proc. Symposium on Inequalities*, edited by O. Shisha, pages 205–224, 1967. 15
- [22] James D Murray. *Mathematical Biology. II Spatial Models and Biomedical Applications {Interdisciplinary Applied Mathematics V. 18}*. Springer-Verlag New York Incorporated, 2001. 1
- [23] Rolf Neuhaus. Computation of real radicals of polynomial idealsii. *Journal of Pure and Applied Algebra*, 124(1):261–280, 1998. 8
- [24] G. Pataki. Strong duality in conic linear programming: facial reduction and extended duals. In David Bailey, Heinz H. Bauschke, Frank Garvan, Michel Thera, Jon D. Vanderwerff, and Henry Wolkowicz, editors, *Computational and analytical mathematics*, volume 50 of *Springer Proc. Math. Stat.*, pages 613–634. Springer, New York, 2013. 12, 13
- [25] G. Reid, F. Wang, H. Wolkowicz, and W. Wu. Semidefinite Programming and facial reduction for Systems of Polynomial Equations. *Preprint arXiv:1504.00931v1*, 2015. 1, 93, 95, 96, 108, 112, 120

- [26] A.J. Sommese and C.W. Wampler. *The Numerical solution of systems of polynomials arising in engineering and science*, volume 99. World Scientific, 2005. 1, 4, 25, 53, 93
- [27] F. Sottile. *Real solutions to equations from geometry*, volume 57 of *University Lecture Series*. American Mathematical Society, Providence, RI, 2011. 8, 53, 54, 55, 56, 84, 93, 94, 120
- [28] Silke J Spang. *On the computation of the real radical*. PhD thesis, Thesis, Technische Universität Kaiserslautern, 2007. 7
- [29] Levent Tunçel. *Polyhedral and semidefinite programming methods in combinatorial optimization*, volume 27 of *Fields Institute Monographs*. American Mathematical Society, Providence, RI, 2010. 13
- [30] Jan Verschelde. Algorithm 795: Phcpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software (TOMS)*, 25(2):251–276, 1999. 4
- [31] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of semidefinite programming*. International Series in Operations Research & Management Science, 27. Kluwer Academic Publishers, Boston, MA, 2000. Theory, algorithms, and applications. 10, 11, 12, 68
- [32] W. Wu and G.J. Reid. Finding points on real solution components and applications to differential polynomial systems. In *Proceedings of the 38th international symposium on International symposium on symbolic and algebraic computation*, pages 339–346. ACM, 2013. 5, 26, 83, 85, 121

## Chapter 2

# Geometric involutive bases for positive dimensional polynomial ideals and SDP methods

### 2.1 Introduction

This paper is part of a stream devoted to developing symbolic-numeric prolongation projection algorithms for general systems of partial and differential algebraic equations. Such algorithms prolong (differentiate) such systems and project the prolonged systems to determine obstructions or missing constraints to their integrability. See Kuranishi [18] for proof of termination of such methods using Cartan's geometric involutivity criteria. A by-product of these methods has been their implementation for linear homogeneous partial differential equations with constant coefficients, and consequently for polynomial algebraic systems. See [13] for applications and symbolic algorithms for polynomial systems. The symbolic-numeric version of a geometric involutive form was first described and implemented in Wittkopf and Reid [41]. It was applied to approximate symmetries of differential equations in [6] and to polynomial solving in [32, 31, 35]. See [43] where it is applied to the deflation of multiplicities in multivariate polynomial solving.

The current paper is focused on further development of our geometric involutive basis al-

gorithm particularly in the positive dimensional case, and also in relation to real solving. It is especially motivated by remarkable recent developments concerning real solution of such systems by Lasserre, Laurent and Rostalski [19] and their use of aspects of our prolongation projection algorithm in the paper “*A prolongation-projection algorithm for computing the finite real variety of an ideal*”. They developed a new approach for computing the real radical of zero dimensional polynomial systems using semi-definite programming (SDP) techniques. See [10] for early fundamental work on such problems. Zero dimensional systems are those having finitely many real solutions, and the real radical is the set of polynomials which vanish on these solutions. In contrast to the input systems the output radical systems from their approach are multiplicity free and so are better conditioned for numerical solution techniques. The output radical systems only have real roots and no complex roots. This leads to possibility of lower complexity methods, since current methods for finding real solutions, mostly explicitly, or implicitly pass through complex root formulations. Given the widespread popularity of linear programming (and by implication) SDP methods, the surprising links between this area also open interesting research possibilities. See [4] for a recent book on the connections between semi-definite optimization and convex algebraic geometry.

We briefly list some background references. There have been considerable recent advances in numerical complex geometry. See especially the books [38, 2] and the references therein. In approaches based on homotopy continuation, positive dimensional components characterize the variety over  $\mathbb{C}$  by certain witness points cut out by intersections of the components with random linear spaces. For a modern text with many references on computational real algebraic geometry see [1]. Real algebraic geometry is a vast subject with many applications. Sturm’s ancient method on counting real roots of a polynomial in an interval is central to Tarski’s real quantifier elimination [40] and was further developed by Seidenberg [36]. One of the most important algorithms of real algebraic geometry is cylindrical algebraic decomposition. CAD was introduced by Collins [9] and improved by Hong [17] who made Tarski’s quantifier elimination algorithmic. This algorithm decomposes  $\mathbb{R}^n$  into cells on which each polynomial of a given system has constant sign. The projections of two cells in  $\mathbb{R}^n$  to  $\mathbb{R}^k$  with  $k < n$  either don’t intersect or are equal. The computational cost of this algorithm, which is doubly exponential [11], is a major barrier to its application. See [8] and [7] for modern improvements

using triangular decompositions. For approaches based on obtaining witness points for the real positive dimensional case see [34, 15, 16, 42]. Homotopy methods are used in [21] and [3] for real algebraic geometry. Recently such moment matrix completion techniques are explored by Zhi et al in [22] for finding at least one real root of a given semi-algebraic system. Furthermore, based on critical point technique and moment matrix completion, they studied the computation of verified real solutions on positive dimensional system in [44].

As part of our initial exploration of this area, in this paper, we make some improvements in our geometric involutive bases, by enabling the explicit extraction of projected systems and hence reducing the size of matrices that can appear in intermediate computations. Similarly motivated by the extension of these methods to the positive dimensional case we explore the interplay between geometric involutive bases and the new SDP methods. The symbol space of a polynomial system or kernel of the matrix of its highest coefficients is the geometric generalization of the highest coefficient of a polynomial. Certain projections within the symbol space encode a geometric test - an analogue of the S-polynomials in Gröbner basis approaches - for new members of the polynomial ideal. We provide details and example of this in the numerical case. An attempt in this paper is made to minimize use of terminology from the jet geometry of partial differential equations, in order to make this accessible to a wider audience.

## 2.2 Brief background on ideals and varieties

In this section we briefly sketch some basic objects from real and complex algebraic geometry and introduce some notation for our paper.

### 2.2.1 Some basic objects in complex algebraic geometry

Consider the set  $\mathbb{C}[x_1, x_2, \dots, x_n]$  of multivariate polynomials with complex coefficients in the complex variables  $x = (x_1, x_2, \dots, x_n) \in \mathbb{C}^n$ . Then  $\mathbb{C}[x_1, x_2, \dots, x_n]$  is a ring. Given  $P = \{p_1(x), p_2(x), \dots, p_m(x)\} \subseteq \mathbb{C}[x_1, x_2, \dots, x_n] = \mathbb{C}[x]$  its solution set or variety is:

$$V_{\mathbb{C}}(p_1, p_2, \dots, p_m) = \{x \in \mathbb{C}^n : p_j(x) = 0, 1 \leq j \leq m\} \quad (2.1)$$

For brevity we sometimes write  $V_{\mathbb{C}}(P) = \{x \in \mathbb{C}^n : P(x) = 0\}$ . Upper case letters  $P, Q, R$ , etc will denote sets of polynomials and lower case letters  $p, q$  etc will denote individual polynomials.

The ideal over  $\mathbb{C}$  generated by  $P = \{p_1, \dots, p_k\}$  is:

$$\langle P \rangle_{\mathbb{C}} = \langle p_1, \dots, p_k \rangle_{\mathbb{C}} = \{f_1 p_1 + \dots + f_k p_k : f_j \in \mathbb{C}[x], 1 \leq j \leq k\} \quad (2.2)$$

and its associated radical ideal over  $\mathbb{C}$  is

$$\begin{aligned} \sqrt[\mathbb{C}]{\langle P \rangle_{\mathbb{C}}} &= \{f \in \mathbb{C}[x] : f(x) = 0 \text{ for all } x \in V_{\mathbb{C}}(P)\} \\ &= \{f \in \mathbb{C}[x] : f^m \in \langle P \rangle_{\mathbb{C}} \text{ for some } m \in \mathbb{N}\} \end{aligned} \quad (2.3)$$

where  $\mathbb{N}$  is the set of non-negative integers.

**Example 2.2.1** *To make this paper accessible to a wide audience we illustrate first some of the main ideas on the simple and well-known case of systems of univariate polynomials. Given a system of  $k$  univariate polynomials  $P = \{p_1, \dots, p_k\}$  with coefficients from some computable field (e.g.  $\mathbb{Q}$ ), a Gröbner basis (or gcd) computation returns a single polynomial  $q(x)$ :*

$$\langle q \rangle_{\mathbb{C}} = \langle p_1, \dots, p_k \rangle_{\mathbb{C}} \quad (2.4)$$

*The factorization of  $q(x)$  over  $\mathbb{C}$  has form:*

$$q(x) = a(x - a_1)^{n_1} \dots (x - a_\ell)^{n_\ell} \quad (2.5)$$

*where the roots  $a_j \in \mathbb{C}$  of  $q(x)$  are distinct. Though the  $a_j$  can't be found in general by finitely many rational operations the so-called square-free factorization can be found by such operations yielding:*

$$\tilde{q}(x) = \frac{q(x)}{\gcd(q(x), q'(x))} = a(x - a_1) \dots (x - a_\ell) \quad (2.6)$$

*For this example the ideal, variety and radical ideal over  $\mathbb{C}$  are:*

$$\begin{aligned} \langle P \rangle_{\mathbb{C}} &= \{g(x) \cdot (x - a_1)^{n_1} \dots (x - a_\ell)^{n_\ell} : g(x) \in \mathbb{C}[x]\} \\ V_{\mathbb{C}}(P) &= \{a_1, a_2, \dots, a_\ell\} \\ \sqrt[\mathbb{C}]{\langle P \rangle_{\mathbb{C}}} &= \{g(x) \cdot (x - a_1) \dots (x - a_\ell) : g(x) \in \mathbb{C}[x]\} \end{aligned} \quad (2.7)$$

*For sophisticated generalizations to primary decomposition for multivariate systems see Gianni et al. [14].*

### 2.2.2 Some basic objects in real algebraic geometry

Suppose that  $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$  and consider a system of  $k$  multivariate polynomials  $P = \{p_1(x), p_2(x), \dots, p_k(x)\} \subseteq \mathbb{R}[x_1, x_2, \dots, x_n]$  with real coefficients. Its solution set or variety is

$$V_{\mathbb{R}}(p_1, \dots, p_k) = \{x \in \mathbb{R}^n : p_j(x) = 0, 1 \leq j \leq k\} \quad (2.8)$$

The ideal generated by  $P = \{p_1, \dots, p_k\} \subseteq \mathbb{R}$  is:

$$\langle P \rangle_{\mathbb{R}} = \langle p_1, \dots, p_k \rangle_{\mathbb{R}} = \{f_1 p_1 + \dots + f_k p_k : f_j \in \mathbb{R}[x], 1 \leq j \leq k\} \quad (2.9)$$

and its associated radical ideal over  $\mathbb{R}$  is defined as

$$\sqrt[\mathbb{R}]{\langle P \rangle_{\mathbb{R}}} = \{f \in \mathbb{R}[x] : f^{2m} + \sum_{j=1}^s q_j^2 \in \langle P \rangle_{\mathbb{R}} \text{ for some } q_j \in \mathbb{R}[x], m \in \mathbb{N} \setminus \{0\}\} \quad (2.10)$$

A fundamental result [5] (originally proved in [33]) is:

**Theorem 2.2.1** [*Real Nullstellensatz*] For any ideal  $I \subseteq \mathbb{R}[x]$  we have  $\sqrt[\mathbb{R}]{I} = I(V_{\mathbb{R}}(I))$ .

Consequently

$$\sqrt[\mathbb{R}]{\langle P \rangle_{\mathbb{R}}} = \{f(x) \in \mathbb{R}[x] : f(x) = 0 \text{ for all } x \in V_{\mathbb{R}}(P)\} \quad (2.11)$$

**Remark** An ideal  $I \subseteq \mathbb{R}[x]$  is real radical if and only if for all  $p_1, \dots, p_k \in R[x]$ :

$$p_1^2 + \dots + p_k^2 \in I \implies p_1, \dots, p_k \in I. \quad (2.12)$$

For these and many other results see [1] and the references cited therein.

**Example 2.2.2** Consider the simplest case of a system of  $k$  univariate polynomials in some computable subfield of  $\mathbb{R}$  (e.g.  $\mathbb{Q}$ ). Then as in the complex case a Gröbner basis of such a system yields a single polynomial  $q(x)$  having the same roots. Discarding the factors with complex roots with nonzero imaginary parts yields a polynomial of form:

$$\tilde{q}(x) = b(x - b_1)^{m_1} \dots (x - b_j)^{m_j} \quad (2.13)$$

where  $b_1, b_2, \dots, b_j$  are the real roots and  $m_1, \dots, m_j$  their corresponding multiplicities. Then

$$\begin{aligned} \langle P \rangle_{\mathbb{R}} &= \{f(x) \cdot (x - b_1)^{m_1} \dots (x - b_j)^{m_j} : f(x) \in \mathbb{R}[x]\} \\ V_{\mathbb{R}}(P) &= \{b_1, b_2, \dots, b_j\} \\ \sqrt[\mathbb{R}]{\langle P \rangle_{\mathbb{R}}} &= \{g(x) \cdot (x - b_1) \dots (x - b_j) : g(x) \in \mathbb{R}[x]\} \end{aligned} \quad (2.14)$$



## 2.3 Geometric prolongation and projection for polynomial systems

In this section we give a brief description of the well-known presentation of polynomial systems as linear functions of their monomials and the related coefficient matrix and its kernel and row space [39, 25, 26, 24] and historical work by Macaulay [23]. We describe a type of elimination called geometric projection and then describe geometric prolongation resulting from multiplying polynomials by monomials.

We exploit the well-known correspondence between polynomial systems and systems of constant coefficient linear homogeneous PDE. This equivalence has been extensively studied and exploited in the exact case by Gerdt [13] and his co-workers in their development of *involution bases*. Our geometric involutive bases are involutive by the geometric criteria in [18, 29, 37] and more distantly related to that of [13] which are closer relatives of Gröbner bases.

Consider a system of  $\ell$  polynomials  $P \subseteq \mathbb{K}[x]$  of degree  $d$  in the variables  $x = (x_1, \dots, x_n)$  where  $\mathbb{K} = \mathbb{R}$  or  $\mathbb{C}$ . Monomials are denoted by  $x^\alpha := x_1^{\alpha_1} \dots x_n^{\alpha_n}$  where  $\alpha \in \mathbb{N}^n$  and the degree of  $x^\alpha$  is  $|\alpha| = \alpha_1 + \dots + \alpha_n$ . Then the system  $P$  can be written as:

$$P = \left\{ \sum_{|\alpha| \leq d} a_{k,\alpha} x^\alpha : k = 1, \dots, \ell \right\} \quad (2.15)$$

To apply the methods of numerical linear algebra the system is converted into matrix form [39, 25, 26, 24].

**Definition 2.3.1 (Coefficient Matrix  $C(P)$ ,  $J^d$  and vector of monomials)** Denote the coefficient matrix of  $P$  in (2.15) by  $C(P)$ . Let  $\mathbf{x}^{(\leq d)}$  be the column vector of monomials  $x^\alpha$  with  $0 \leq |\alpha| \leq d$  sorted by graded reverse lexicographic order. We suppose that the columns of  $C(P)$  are sorted in the same order. Then  $P = C(P)\mathbf{x}^{(\leq d)}$  where  $C(P) \in \mathbb{R}^{\ell \times N(n,d)}$  and  $N(n,d) := \binom{d+n}{d}$  is the number of monomials in  $\mathbf{x}^{(\leq d)}$ . Polynomials can be equivalently represented by the row vectors of  $C(P)$ , that is as vectors in  $J^d := \mathbb{R}^{N(n,d)}$ .

Prolonging polynomials by multiplying them by monomials is an essential geometric operation in this paper.

**Definition 2.3.2 (prolongations  $\widehat{\mathbf{D}}$  and  $\widetilde{\mathbf{D}}$ )** Consider a system of polynomials  $P$  of degree  $d$ . Let  $p \in P$  have degree  $\bar{d}$ . Then the prolongation of  $p$  written  $\widehat{\mathbf{D}}(p)$  is defined as  $\widehat{\mathbf{D}}(p) = \{p\} \cup \{x_j p : 1 \leq j \leq n\}$ . The prolongation of the system  $P$  is defined as  $\widehat{\mathbf{D}}^k(P) = \{x^\alpha p : 0 \leq \deg(x^\alpha p) \leq d + k, \alpha \in \mathbb{N}^n, p \in P\}$ . Equivalently we can represent the prolongation geometrically as the span of the corresponding row vectors of  $C(\widehat{\mathbf{D}}^k P)$ , which we denote by  $\widetilde{\mathbf{D}}^k(P) := \text{rowsp}(C(\widehat{\mathbf{D}}^k P))$  which is a subspace of  $J^{d+k}$ .

**Example 2.3.1** Suppose  $x = (y, z)$  and  $P = \{2, 2y + z\}$ . Then  $\widehat{\mathbf{D}}(P) = \{2, 2y, 2z, 2y^2, 2yz, 2z^2, 2y^2 + yz, 2yz + z^2\}$ .

**Definition 2.3.3 (projections  $\widehat{\pi}$  and  $\widetilde{\pi}$ )** Consider a polynomial system of degree  $d \geq 1$  written in the form  $P = C(P)\mathbf{x}^{(\leq d)}$  with the columns of  $C(P)$  sorted in descending order by degree. The rows in the Gauss echelon form of  $C(P)$  with pivots of degree less than  $d$  span a subspace of  $J^{d-1}$  which we denote by  $\widetilde{\pi}(P)$ . We denote the set of polynomials of degree  $\leq d - 1$  corresponding to the row vectors by  $\widehat{\pi}(P)$ . Iterations of projections  $\widehat{\pi}^\ell(P) \subset \mathbb{R}[x]$  and equivalently  $\widetilde{\pi}^\ell(P) \subset J^{d-\ell}$  are defined similarly.

We have adopted an abbreviated notation for prolongation and projection here to avoid cumbersome indices indicating the spaces on which these operators act. We will also need to prolong and project kernels of the coefficient matrices of polynomial systems.

**Definition 2.3.4 (prolongation  $\mathbf{D}$  and projection  $\pi$  on the kernel)** Consider a polynomial system  $P \subset \mathbb{R}[x]$  of degree  $d$ . Given a subspace  $V$  of  $J^d$  and  $\ell \leq d$  define  $\pi^\ell(V)$  as the vectors of  $V$  with the components of degree  $\geq d - \ell$  discarded. To abbreviate notation we will write  $\pi^\ell(P) := \pi^\ell \ker C(P)$ . The  $k$ -th prolongation of the kernel is  $\mathbf{D}^k(P) := \ker C(\widehat{\mathbf{D}}^k P)$ .

In summary we have presented three (!) notations for prolongation and projection since we need to work directly with them sometimes as polynomial systems, and sometimes row spaces or kernels. The row space and kernel are orthogonal to each other in  $J^d$ . Projection in the kernel is the usual projection operator  $\pi^\ell$ . Geometrically the corresponding projection in the row space can be obtained as the orthogonal complement of  $\pi^\ell(P)$ . Alternatively it can be obtained by first considering  $J^{d-\ell}$  as a subspace in  $J^d$  and then intersecting the subspace  $J^{d-\ell}$  with  $\text{rowsp}(P)$ .

Suppose that  $A = C(P)$  is the coefficient matrix of a system of polynomials  $P$ . To numerically implement an approximate involutive form method, we proposed in [6, 41, 31] a numeric version of the projection operator based on singular value decomposition (SVD). We first find the SVD of  $A$  given by  $A = U \cdot \Sigma \cdot V$  where  $U$  and  $V$  are unitary matrices and  $\Sigma$  is a diagonal matrix whose diagonal entries are real decreasing non-negative numbers. The approximate rank  $r$  is the number of singular values bigger than a fixed tolerance. Deleting the first  $r$  rows of  $V$  yields an approximate basis for  $\ker A$  and an estimate for  $\dim \ker A$ . Deleting highest degree components of the vectors in this basis, yields an approximate spanning set for  $\pi \ker A$  and an estimate for  $\dim \pi \ker A$ . If desired further computation yields bases for  $\pi \ker A$ . Then we compute the kernel of the spanning set of  $\pi \ker A$ . Similarly we can compute approximate spanning sets and if desired bases of the prolongations and projections of the system.

**Remark 2.3.1 (Alternative representations and extraction of intermediate systems)** *In summary prolongation and projection can equivalently be computed in either the kernel or the rowspace, and at any time polynomial generators can be extracted. Underlying this is a 1 to 1 correspondence between vector spaces (not elements): in particular between the row spaces and its orthogonal complement, the kernel.*

**Example 2.3.2** *Consider*

$$P = \{x^8 - x^4 - 2, x^8 - 3x^4 + 2\} \subseteq \mathbb{R}[x] \quad (2.16)$$

Here the coefficient matrix is given by  $C(P)$  below:

$$C(P) \cdot \mathbf{x}^{(\leq 8)} = \begin{pmatrix} -2 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 & -3 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} 1 \\ x^1 \\ \vdots \\ x^7 \\ x^8 \end{bmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (2.17)$$

The most familiar computation for most readers is to eliminate the polynomials as in a Gröbner basis calculation:  $x^8 - x^4 - 2 - (x^8 - 3x^4 + 2) = 2x^4 - 4$ . This can also be done as a computation on the row space of  $C(P)$ , yielding the result as the generator of  $\tilde{\pi}^4 P$ . Equivalently by Remark 2.3.1 we can compute the result by projecting basis vectors of the kernel of  $C(P)$  obtaining  $\pi^4 P$

	$k = 0$	$k = 1$	$k = 2$	$k=3$
$\ell = 0$	7	6	5	4
$\ell = 1$	7	6	5	4
$\ell = 2$	6	6	5	4
$\ell = 3$	5	5	5	4
$\ell = 4$	4	4	4	4
$\ell = 5$	4	4	4	4
$\ell = 6$	3	4	4	4
$\ell = 7$	2	3	4	4
$\ell = 8$	1	2	3	4
$\ell = 9$		1	2	3
$\ell = 10$			1	2
$\ell = 11$				1

Figure 2.1: Table of  $\dim \pi^\ell \mathbf{D}^k P$  for (2.17) for Example 2.3.2. The (red) boxed 4 in the first column corresponds to  $\pi^4 P$  and a geometric involutive basis for  $P$  as found by Algorithm 2.4.1. The blue and black boxed 4's in the fourth column correspond to geometric involutive bases for  $P$ .

and then recover the generator  $2x^4 - 4$ . The original 8 degree polynomials can be discarded since they are multiples of  $2x^4 - 4$ . In particular a Gröbner basis for the ideal generated by  $P$  is

$$x^4 - 2 \tag{2.18}$$

The kernel of  $C(P)$  is easily calculated numerically by the SVD. We obtain the table of dimensions for the projections of  $\ker C(P)$  in Figure 2.1. We use singular value decomposition to compute its kernel and then project its vectors to  $\pi^4 P$ . The generator corresponding to this projection is:

$$0.4472136 x^4 - 0.8944272. \tag{2.19}$$

where the coefficients here and elsewhere in the paper have been truncated from 15 digits to 7 digits. After normalization, we get the generator  $x^4 - 2$ .

## 2.4 Geometric involutive bases

In this section we describe the geometric involutive form of a polynomial system. For a more detailed description see [6, 30, 41, 31].

Exact elimination methods for exactly given polynomial systems (e.g. Gröbner Bases), usually employ Gaussian Elimination (e.g. linear elimination of monomials). Such exact methods usually depend on the ordering of input (e.g. term ordering in the case of Gröbner Bases), and so are coordinate dependent. Since the order of elimination can force division by small leading entries, such methods are generally unstable, when used on approximate systems. In contrast, exact elimination methods from the geometric theory of PDE are coordinate independent [18, 29] and this motivated our study of numerical versions of such methods which is continued in this paper.

### 2.4.1 Symbol, class and Cartan involution test

**Definition 2.4.1 (Symbol matrix and class of a monomial)** *Given a polynomial system of degree  $d$ , its symbol matrix, denoted  $S(P)$  is the submatrix of  $C(P)$  corresponding to its degree  $d$  monomials. Consider a monomial  $x^\alpha$  where  $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$ . Then the class of  $x^\alpha$  is the least  $j$  such that  $\alpha_j \neq 0$ .*

For Example 2.3.2 the symbol matrix is the submatrix  $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$  of  $C(P)$  given in (2.17). Consider the system

$$P = \{x_2^2 - 1, 2x_1x_2 - 3x_1\} \quad (2.20)$$

For what follows we sort the columns of the symbol matrix in descending order according to class. The degree two monomials are  $x^{(0,2)} = x_2^2$ ,  $x^{(1,1)} = x_1x_2$ ,  $x^{(2,0)} = x_1^2$ . Here  $x_2^2$  is class 2. Monomials  $x_1x_2$  and  $x_1^2$  are class 1. Then the symbol matrix is:

$$S(P) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix} \quad (2.21)$$

**Definition 2.4.2 (Cartan test for involutivity of the Symbol)** *Suppose that the columns of the symbol matrix for a system of degree  $d$  are sorted in descending order by class and that it is*

reduced to Gauss echelon form. For  $k = 1, 2, \dots, n$  define the quantities  $\beta_d^{(k)}$  as the number of pivots in this reduced matrix of class  $k$ . Then in a generic system of coordinates the symbol is involutive if:

$$\sum_{k=1}^{k=n} k\beta_d^{(k)} = \text{rank } \mathcal{S}(\widehat{\mathbf{D}}P) \quad (2.22)$$

The following combinatorial quantities will be useful in our numerical determination of involutivity of symbol matrices. Consider systems in  $n$  variables of degree  $d$ .

Denote:

$$\begin{aligned} N(n, d) &= \binom{n+d}{d} = \text{Number of monomials of degree } \leq d \\ N_{\deg}(n, d) &= \binom{n+d-1}{d} = \text{Number of monomials of degree } d \\ N_c(n, d, k) &= \binom{n+d-k-1}{d-1} = \text{Number of class } k \text{ monomials of degree } d \end{aligned} \quad (2.23)$$

**Example 2.4.1** For system  $P$  given in (2.20):

$$N(2, 2) = 6, N_{\deg}(2, 2) = 3, N_c(2, 2, 1) = 2, N_c(2, 2, 2) = 1 \quad (2.24)$$

The symbol matrix (2.21) is already in Gauss echelon form with respect to class. There is one pivot of class 2 so  $\beta_2^{(2)} = 1$  and one pivot of class 1 so  $\beta_2^{(1)} = 1$ . Also an easy calculation gives  $\text{rank } \mathcal{S}(\widehat{\mathbf{D}}P) = 3$ . So

$$\sum_{k=1}^{k=2} k\beta_d^{(k)} = 3 = \text{rank } \mathcal{S}(\widehat{\mathbf{D}}P) \quad (2.25)$$

and the symbol is involutive. In all cases  $\sum_{k=1}^{k=n} k\beta_d^{(k)} \leq \text{rank } \mathcal{S}(\widehat{\mathbf{D}}P)$ . Indeed in our example if we reverse the order of the coordinates and recalculate we get  $\mathcal{S}(P) = \begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ . Then  $\beta_2^{(2)} = 0, \beta_2^{(1)} = 2$  and  $\sum_{k=1}^{k=2} k\beta_d^{(k)} = 2 < \text{rank } \mathcal{S}(\widehat{\mathbf{D}}P)$  so the test indicates a non-involutive symbol however the result may be due to the coordinates being nongeneric which is indeed the case here. A generic linear change of coordinates by a random  $2 \times 2$  matrix then shows the symbol is involutive.

To extract a matrix for the symbol space of the variables of degree  $d$  we proceed as follows for a system  $P$  of degree  $d' \geq d$ . Suppose that vectors that are a basis for the kernel of  $C(P)$  form the rows of a matrix  $B$ . First numerically project the kernel of the system  $P$  onto the subspace  $J^d$  via  $\pi^{d'-d}P$  by deleting the coordinates in the basis of degree  $> d$  to obtain for  $\pi^{d'-d}P$  a spanning set  $\tilde{B}$  given by the remaining rows of  $B$ . Then delete the columns in  $\tilde{B}$  corresponding to variables of degree  $< d$  to obtain a matrix  $A_d$  corresponding to the orthogonal complement of the symbol for degree  $d$ . Let  $A_d^{(k)}$  be the submatrix of  $\tilde{B}$  with columns corresponding to class  $k$  or less deleted. In generic coordinates

$$\beta_d^{(k)} = N_c(n, d, k) - (\text{rank } A_d^{(k-1)} - \text{rank } A_d^{(k)}), \quad k = 1 \dots n. \quad (2.26)$$

Then the SVD can approximate the ranks in this equation for carrying out the Cartan Test (2.22).

**Definition 2.4.3 (Involutive System)** *A system of polynomials  $P \in \mathbb{R}[x]$  is involutive if  $\dim \pi DP = \dim P$  and the symbol of  $P$  is involutive.*

**Definition 2.4.4 (Projected Involutive System)** *Consider a system of polynomials  $P \in \mathbb{R}[x]$  of degree  $d$ . Suppose that  $k, \ell$  are integers with  $k \geq 0$  and  $0 \leq \ell \leq k + d$ . Then  $\pi^\ell D^k P$  is projectively involutive at prolongation order  $k$  and projected order  $\ell$ , if  $\pi^\ell D^k P$  satisfies the projected elimination test*

$$\dim \pi^\ell D^k P = \dim \pi^{\ell+1} D^{k+1} P \quad (2.27)$$

*and the symbol of  $\pi^\ell D^k P$  is involutive.*

In [6] it is proved:

**Theorem 2.4.1** *A system is projectively involutive if and only if it is involutive.*

**Theorem 2.4.2 (Criterion for zero dimensional involutive system)** *A zero dimensional system of polynomials  $P \in \mathbb{R}[x]$  is projectively involutive at order  $k$  and projected order  $\ell$  if and only if  $\pi^\ell D^k P$  satisfies the projected elimination test (2.27) and*

$$\dim \pi^\ell D^k P = \dim \pi^{\ell+1} D^{k+1} P \quad (2.28)$$

This criterion is used by Lasserre et al [20] in their prolongation projection algorithm to determine the finite real radical. When there are 2 variables then it is easily shown that:

$$S \pi^\ell \mathbf{D}^k P \text{ is involutive} \iff \dim S \pi^\ell \mathbf{D}^{k+1} P = \dim S \pi^\ell \mathbf{D}^k P \quad (2.29)$$

and this gives a computationally easy characterization by using

$$\dim S \pi^\ell \mathbf{D}^k P = \dim \pi^\ell \mathbf{D}^k P - \dim \pi^{\ell+1} \mathbf{D}^k P \quad (2.30)$$

The criterion in (2.27) applies to both zero and positive dimensional bivariate systems.

### 2.4.2 Projected involutive form algorithm

The following method completes systems to approximate involutive form. We seek the smallest  $k$  such that there exists an  $\ell$  with  $\pi^\ell \mathbf{D}^k P$  approximately involutive, and generates the same ideal as the input system. We choose the system corresponding to the largest such  $\ell \leq k$  if there are several such values for the given  $k$ .



---

**Algorithm 2.4.1:** Projected involutive basis

---

**Input:**  $Q \subseteq \mathbb{R}[x_1, \dots, x_n]$ . A tolerance  $\epsilon$ .Set  $k := 0$ ,  $d := \deg(Q)$  and  $P := \ker C(Q)$ **repeat**    Compute  $\mathbf{D}^k(P)$     Initialize set of involutive systems  $I := \{\}$     **for**  $\ell = 0 \cdots (d + k)$  **do**        Compute  $R := \pi^\ell \mathbf{D}^k(P)$         **if**  $R$  involutive **then**  $I := I \cup \{R\}$  **end if**    **end do**    Remove systems  $\bar{R}$  from  $I$  not satisfying  $\mathbf{D}^{d+k-\bar{d}}\bar{R} \subseteq \mathbf{D}^k(P)$  where  $\bar{d}$  is the degree of  $\bar{R}$ .     $k := k + 1$ **until**  $I \neq \{\}$ **Output:** Return the polynomial generators of the involutive system  $\bar{R}$  in  $I$   
of lowest degree  $\bar{d}$ .

---

Note that this algorithm works on kernels, but could by Remark 2.3.1 equivalently work on their orthogonal complements – the associated row spaces. The condition  $\mathbf{D}^{d+k-\bar{d}}\bar{R} \subseteq \mathbf{D}^k(P)$  is a standard subspace inclusion test for the prolonged kernels. It ensures that the output system generates the same ideal as the input system and has the same solutions.

**Decreasing degrees by extracting involutive projections**

We note that a simple illustration of Algorithm 2.4.1 is Example 2.3.2 where all univariate polynomials are involutive. This algorithm is an improvement on that published in [35] where to ensure the inclusion conditions for positive dimensional ideals, the number of projections was limited to  $0 \leq \ell \leq k$ . So Algorithm 2.4.1 can return generators of lower degree than the algorithm published in [35].

## 2.5 Moment matrices and SDP

### 2.5.1 Moment Matrices

Here we focus just on the construction of moment matrices. For the theoretical background the reader is directed to [20].

A moment matrix is a symmetric matrix  $M = (M_{\alpha,\beta})$  indexed by  $\mathbb{N}^n$  ( $\alpha, \beta \in \mathbb{N}^n$ ). Here  $\alpha$  is the index for rows,  $\beta$  is the index for columns. Without loss  $M_{0,0} = 1$ .

Given a multivariate polynomial system  $P \subseteq \mathbb{R}[x_1, \dots, x_n]$ . Let  $d = \deg(P)$  and  $M \in \mathbb{R}^{N(n,d) \times N(n,d)}$  be the truncated moment matrix. The linear constraints imposed by  $P$  are constructed as

$$M \cdot A^T = 0; A = C(\widehat{\mathbf{D}}^d(P)), \quad (2.31)$$

where  $C$  is the coefficient matrix function given in Definition 3.2.1.

### 2.5.2 Moment matrix for univariate example

In Example 2.3.2 a degree 8 input system was reduced to a degree 4 output polynomial  $p = x^4 - 2$ . Then in matrix form the polynomial is

$$Bv = \begin{pmatrix} -2 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = 0, \ker B = \text{span}_{\mathbb{R}} \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right\} \quad (2.32)$$

The moment matrix is the infinite matrix whose  $(\alpha, \beta)$  entry is  $u_{\alpha+\beta}$  and  $\alpha, \beta \in \mathbb{N}^n$  given by:

$$M = \begin{pmatrix} u_0 & u_1 & u_2 & u_3 & u_4 & \cdots \\ u_1 & u_2 & u_3 & u_4 & u_5 & \cdots \\ u_2 & u_3 & u_4 & u_5 & u_6 & \cdots \\ u_3 & u_4 & u_5 & u_6 & u_7 & \cdots \\ u_4 & u_5 & u_6 & u_7 & u_8 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (2.33)$$

In the SDP-moment matrix approach the given polynomial system, in this case  $\{x^4 - 2\}$ , is first prolonged to twice its degree:

$$\widehat{\mathbf{D}}^4\{x^4 - 2\} = \{x^4 - 2, x^5 - 2x, x^6 - 2x^2, x^7 - 2x^3, x^8 - 2x^4\} \quad (2.34)$$

From (2.31) the constraint system when we impose  $u_0 = 1$  is equivalent to the linear system

$$u_4 - 2 = 0, u_5 - 2u_1 = 0, u_6 - 2u_2 = 0, u_7 - 2u_3 = 0, u_8 - 2u_4 = 0 \quad (2.35)$$

which can be regarded as the rewrite rules:  $u_4 \rightarrow 2, u_5 \rightarrow 2u_1, u_6 \rightarrow 2u_2, u_7 \rightarrow 2u_3, u_8 \rightarrow 2u_4 \rightarrow 4$ . Imposing these constraints the truncated moment matrix to degree 8 is

$$M = \begin{pmatrix} 1 & u_1 & u_2 & u_3 & 2 \\ u_1 & u_2 & u_3 & 2 & 2u_1 \\ u_2 & u_3 & 2 & 2u_1 & 2u_2 \\ u_3 & 2 & 2u_1 & 2u_2 & 2u_3 \\ 2 & 2u_1 & 2u_2 & 2u_3 & 4 \end{pmatrix} \quad (2.36)$$

The moment matrix (2.36) is then sent to the SDP solver Yalmip in Matlab to numerically compute a generic point  $(u_1, u_2, u_3)$  if possible such that  $M$  is a positive semidefinite matrix with maximum rank. This solver returns an approximation which can be recognized for illustrative convenience as  $(u_1, u_2, u_3) = (0, \sqrt{2}, 0)$ . Its associated moment matrix and moment matrix

kernel are:

$$M = \begin{pmatrix} 1 & 0 & \sqrt{2} & 0 & 2 \\ 0 & \sqrt{2} & 0 & 2 & 0 \\ \sqrt{2} & 0 & 2 & 0 & 2\sqrt{2} \\ 0 & 2 & 0 & 2\sqrt{2} & 0 \\ 2 & 0 & 2\sqrt{2} & 0 & 4 \end{pmatrix}, \ker M = \text{span}_{\mathbb{R}} \left\{ \begin{pmatrix} -2 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -\sqrt{2} \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -\sqrt{2} \\ 0 \\ 1 \\ 0 \end{pmatrix} \right\} \quad (2.37)$$

The kernel corresponds to the generating set

$$\{\sqrt{2} - x^2, 2 - x^4, \sqrt{2}x - x^3\} \quad (2.38)$$

Applying geometric involutive form algorithm yields a geometric involutive basis

$$\{\sqrt{2} - x^2\} \quad (2.39)$$

The last two polynomials are a consequence of  $\sqrt{2} - x^2$  by our inclusion test, so are discarded.

By Laurent and Rostalski [20], this is a basis of the real radical.

## 2.6 Combining geometric involutive bases and moment matrix methods

### 2.6.1 Geometric involutive form and moment matrix algorithms

In this section we outline algorithms for combining geometric involutive form and moment matrix methods.

**Proof of the termination of Algorithm 2.6.1:** We prove termination of the GIF– M Method under the assumption that suitable generic points, if available, are determined at each iteration of the method.

**Rank-Dim-Involutive Stopping Criterion:** A natural termination criterion used in Algorithm 2.6.1 is that the generators stabilize at some iteration and the system is involutive:

$$\text{gen}(\text{GIF}(Q)) = \text{gen}(\ker \mathbf{M}(Q)) \text{ and } Q \text{ involutive} \quad (2.40)$$

**Algorithm 2.6.1:** GIF– M Method**Input:**  $P = \{p_1, \dots, p_k\} \subseteq \mathbb{R}[x_1, \dots, x_n]$  $Q_0 := P$  $j := 0$ **do** $d := \dim \ker \text{GIF}(Q_j)$  $Q_{j+1} := \text{gen}(\text{GIF}(Q_j))$  $r := \text{rank}(\mathbb{M}(Q_{j+1}))$  $Q_{j+2} := \text{gen}(\ker \mathbb{M}(Q_{j+1}))$  $j := j + 2$ **until**  $r = d$ **Output:**  $Q = \{q_1, \dots, q_\ell\} \subseteq \mathbb{R}[x_1, \dots, x_n]$  $Q$  is in geometric involutive form

$$\sqrt[\mathbb{R}]{\langle P \rangle_{\mathbb{R}}} \supseteq \langle Q \rangle_{\mathbb{R}} \supseteq \langle P \rangle_{\mathbb{R}}.$$

Since different representations of the rings are involved we will focus on one, that of polynomial generators during the proof.

In terms of generators our termination criterion  $\text{rank}(\mathbb{M}(Q_{j+1})) = \dim \ker \text{GIF}(Q_j)$  is expressed as  $\text{gen}(\text{GIF}(Q_j)) = \text{gen}(\ker \mathbb{M}(Q_{j+1}))$ .

Then  $\text{gen}(\ker \mathbb{M}(Q_{j+1}))$  and  $\text{gen}(\text{GIF}(Q_j))$  are both ideals of the system  $P$ . Since a generator of the geometric involutive form will also be a generator of the ideal in the moment matrix at each iteration we have  $\text{gen}(\text{GIF}(Q_j)) \subseteq \text{gen}(\ker \mathbb{M}(Q_{j+1}))$  in our algorithm. Suppose the algorithm never stops, then we will get a infinite ascending chain of ideals with a strict inclusion at each iteration of the form  $Q_j \subset Q_{j+1}$  where  $Q_j = \text{gen}(\text{GIF}(Q_{j-1}))$  and  $Q_{j+1} = \text{gen}(\ker \mathbb{M}(Q_j))$ . This is a violation of the ascending chain condition since  $\mathbb{R}[x_1, \dots, x_n]$  is a Noetherian Ring. Therefore, the generators must stabilize in the end and when stabilized,  $Q$  is also involutive.  $\square$

The algorithm above uses the following subroutines.

Note the algorithm 2.4.1 is an explicit implementation of GIF.

---

**Algorithm 2.6.2:** GIF

---

1

**Input:**  $Q \subseteq \mathbb{R}[x_1, \dots, x_n]$ **Output:** Return a geometric involutive form  $\text{GIF}(Q)$ .

---

---

**Algorithm 2.6.3:** M

---

1

**Input:**  $Q \subseteq \mathbb{R}[x_1, \dots, x_n]$ . Set  $d := \deg(Q)$ .

1. Construct the general  $N(n, d) \times N(n, d)$  moment matrix.
2. Construct the involutive prolongation  $D^d Q$ .
3. Use SDP methods to numerically solve for a generic point that maximizes the rank of the moment matrix subject to the constraints  $D^d Q$ .

**Output:** Return  $\mathbb{M}(Q) \geq 0$  the moment matrix evaluated at this generic point.

---

---

**Algorithm 2.6.4:** gen

---

1

**Input:**  $\text{GIF}(Q)$  or  $\ker \mathbb{M}(Q)$ **Output:** Polynomial generators corresponding to  $\text{GIF}(Q)$  or  $\ker \mathbb{M}(Q)$ 

---

## 2.6.2 Two variable example

Consider the polynomial system with two variables  $x$  and  $y$ .

$$P_2 = \{(y^2 - 1)^2, (y^2 - 1)(x^2 - 1)\} \quad (2.41)$$

First we apply GIF to  $P_2$  to compute the involutive form of it. The dimension table is in Figure 2.2.

Now  $\dim \pi^2 \mathbf{D}^2(P_2) = \dim \pi^3 \mathbf{D}^3(P_2)$  so  $\pi^2 \mathbf{D}^2(P_2)$  satisfies one of the conditions for an involutive system. The second condition is that the symbol of  $\pi^2 \mathbf{D}^2(P_2)$  is involutive. Applying the

	$k = 0$	$k = 1$	$k = 2$	$k=3$
$\ell = 0$	13	15	17	19
$\ell = 1$	10	12	14	16
$\ell = 2$	6	9	11	13
$\ell = 3$	3	6	9	11
$\ell = 4$	1	3	6	9

Figure 2.2: Table of  $\dim \pi^\ell \mathbf{D}^k(P_2)$  for system (2.41) The (blue) boxed 11 in the third column corresponds to  $\pi^2 \mathbf{D}^2(P_2)$ .

symbol test (2.29) and we find that  $\dim \mathcal{S} \pi^2 \mathbf{D}^3(P_2) = \dim \mathcal{S} \pi^2 \mathbf{D}^2(P_2) = 2$ , so the symbol of it turns out to be involutive as well. Therefore  $\pi^2 \mathbf{D}^2(P_2)$  is involutive.

Now we apply the subroutine **M** to  $\text{gen}(\pi^2 \mathbf{D}^2(P_2))$  to compute the moment matrix  $M$ . We convert  $\ker M$  into polynomial generators by subroutine **gen**. The dimension of  $\ker M$  is 6 which means there are 6 generators in  $\text{gen}(\ker M)$ , which are moderately complicated numerical polynomials.

We again apply GIF to  $\text{gen}(\ker M)$  to compute the involutive form. The dimension table is shown in Figure 2.3. The input system corresponding to the (red) boxed 9 is already involutive. As mentioned in Remark 2.4.2 in algorithm 2.4.1 and more generally in GIF algorithm, we can extract projected systems of lower degree than input system. This improves on our previous algorithm [35]. We demonstrate this procedure here. In Figure 2.3, the system corresponding to the red boxed 9 is involutive and has degree 4. Since  $N(2, 4) = 15$  there are  $15 - 9 = 6$  polynomials in the system. However descending further down the column of the table, we find the system corresponding to the blue boxed 5 is also involutive. In that case  $N(2, 2) = 6$  so there is only 1 corresponding generator.

We apply **gen** to compute the generator set:

$$\{0.7071067y^2 - 0.7071067 + \text{small terms less than } 10^{-11}\} \quad (2.42)$$

If we apply GIF to equation (2.42), the dimension table is exactly the same as the one in Figure 2.3. Therefore the projected system is equivalent to the input system. After normalization and ignoring small terms, we get  $y^2 - 1$  which is a geometric involutive basis for the real

	$k = 0$	$k = 1$	$k = 2$	$k=3$
$\ell = 0$	9	11	13	15
$\ell = 1$	7	9	11	13
$\ell = 2$	5	7	9	11
$\ell = 3$	3	5	7	9
$\ell = 4$	1	3	5	7

Figure 2.3: Table of  $\dim \pi^\ell \mathbf{D}^k \text{gen}(\ker M)$  for the first GIF–M iteration in Example 2.6.2.

radical for  $P_2$ .

### 2.6.3 Three variable example

In this section we apply the GIF–M method to the following trivariate system with GIF explicitly implemented by Algorithm 2.4.1:

$$P_3 = \begin{cases} x^2y^2 - y^4 + y^2z^2 - x^2 - z^2 + 1 \\ x^2y^2 - y^4 + y^2z^2 + x^2 - 2y^2 + z^2 - 1 \\ x^4z + x^2z^3 - 2x^2y^2 - x^2z - z^3 - 2x^2 + 2y^2 + 2 \\ x^4z + x^2z^3 - 2x^2y^2 + x^2z + z^3 - 2x^2 - 2y^2 - 2 \end{cases} \quad (2.43)$$

We first apply subroutine GIF to  $P_3$ . The dimension table is shown in Figure 2.4.

	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k=4$
$\ell = 0$	46	57	66	73	79
$\ell = 1$	29	38	46	53	59
$\ell = 2$	15	21	27	33	39
$\ell = 3$	9	15	21	27	33
$\ell = 4$	4	9	15	21	27

Figure 2.4: Table of  $\dim \pi^\ell \mathbf{D}^k(P_3)$  for system (2.43).

At prolongation zero of Algorithm 2.4.1 we determine if there are any projected involutive systems whose prolongations yield the same ideal as the system (so that the prolongations can



be discarded). We find such an involutive system  $\pi^2 \mathbf{D}^0(P_3)$  which corresponds to the red boxed 15 in column 1 of Figure 2.4. From the dimension information we can deduce that since the number of monomials of degree  $\leq 3$  is  $N(3, 3) = 20$  there will be  $20 - 15 = 5$  polynomials generators corresponding to  $\pi^2 \mathbf{D}^0(P_3)$ . System  $\pi^3 \mathbf{D}^0(P_3) = \bar{R}$  is of lower degree and also easily found to be involutive. However it does not satisfy the inclusion test of Algorithm 2.4.1 given by  $\mathbf{D}^{d+k-\bar{d}} \bar{R} \subseteq \mathbf{D}^k(P_3)$  which shows that it is not equivalent to the original system. We find that  $\pi^2 \mathbf{D}^0(P_3)$  does satisfy the inclusion output condition, so we exit GIF and apply subroutine  $\mathbb{M}$  to  $\text{gen}(\pi^2 \mathbf{D}^0(P_3))$ . In our previously published method we would have first identified the blue boxed 27 corresponding to the involutive system  $\pi^2 \mathbf{D}^2(P_3)$ . Our approach is a clear improvement, and avoids creating the large degree 5 moment matrix of the previous approach.

We compute the generator set of the moment matrix  $M$  using the subroutine  $\text{gen}(\ker M)$ . The rank of moment matrix is 7 which means  $\text{gen}(\ker M)$  has dimension 13. We apply GIF to  $\text{gen}(\ker M)$  and the dimension table is given in Figure 2.5

	$k = 0$	$k = 1$	$k = 2$	$k = 3$
$\ell = 0$	7	9	11	13
$\ell = 1$	5	7	9	11
$\ell = 2$	3	5	7	9
$\ell = 3$	1	3	5	7

Figure 2.5: Table of  $\dim \pi^\ell \mathbf{D}^k \text{gen}(\ker M)$  in the moment matrix calculation for  $\text{gen}(\pi^2 \mathbf{D}^0(P_3))$ .

In this iteration of GIF three systems are involutive and correspond to the  $\ell = 0, 1, 2$  entries of column 1 of Figure 2.5. Corresponding to the elimination of higher order systems by the inclusion test in Algorithm 2.4.1, we can discard 2 of the 3 systems, which correspond to  $\ell = 0$  and  $\ell = 2$  entries in the first column. The output lower degree geometric involutive basis therefore corresponds to the blue boxed entry in the figure.

At the next iteration the generators corresponding to  $\ell = 1$  are sent to the moment matrix. We find that the termination condition is satisfied, that is  $d = 5 = r$ . The algorithm then terminates with an output of  $10 - 5 = 5$  generators.

To get more insight into the output we now analyze it further. From the Figure 2.5 we see that there is a projected system corresponding to  $\ell = 2$  of dimension 3 and degree 1. When it

is extracted we find a single nice generator:

$$0.8944271 - 0.4472135z + 5.5511151 \times 10^{-17}y \quad (2.44)$$

After dropping off the small term and normalization, we get  $z - 2$ . Now we consider the other generators of degree 2. We can simplify them by substituting  $z = 2$  from the projected generator and find

$$-0.3015113x^2 + 0.3015113y^2 - 0.9045340 + \text{small terms less than } 10^{-15}. \quad (2.45)$$

which is approximately  $x^2 - y^2 + 3$ . Thus our output geometric involutive basis is

$$\{z - 2, x(z - 2), y(z - 2), z(z - 2), x^2 - y^2 + 3\} \quad (2.46)$$

A hand calculation checks that this is a geometric involutive basis for the real radical of the input system.

## 2.7 Discussion

In this paper we present improvements of our numerical geometric involutive bases for polynomial systems of equations. We also began an exploration of the interaction of these methods with SPD programming methods and computation of such bases for positive dimensional real radical ideals.

We give methods to extract and decrease the degree of immediate systems and the output basis. One such tool is an inclusion test whereby higher degree redundant systems can be discarded. Prompted by a number of requests we have given more details of our implementation of Cartan's involutivity test for positive dimensional ideals. Reduction of degree techniques are critical and have been extensively developed in the symbolic case for Gröbner bases [12] and triangular decompositions [7, 8]. Significant progress has also been made in symbolic-numeric methods such as border bases [25, 26, 27, 28] in removing higher degree polynomials. Perhaps the closest objects to geometric involutive bases in the zero dimensional case are H-Bases [24].

Moreover, we were motivated by remarkable recent work by Lasserre and collaborators [20] using SDP methods for identifying the real radical of zero dimensional polynomial ideals.

The work [20] motivated us to combine SDP – moment matrix methods with our geometric involutive bases to approximate positive dimensional real radical ideals. In particular, the termination criterion  $\text{rank}(\mathbf{M}(Q)) = \dim \ker \text{GIF}(Q)$  in Algorithm 2.4.1 is equivalent to the rank stabilization condition in Lasserre [20] for zero dimensional systems. Moreover in our initial explorative experiments we obtained generators for the real radical of positive dimensional ideals for a small set of examples and deserves further study.

In our preliminary study in order to study the interaction between these two methods we focused on an algorithm that cleanly separates the step of taking a geometric involutive basis at each iteration of the algorithm. An alternative strategy that we will pursue in future work is motivated by the approach of Lasserre et al in the zero dimensional case [19]. Instead of demanding a (projected) involutive form at each iteration, they allowed the iteration and prolongation of moment matrices until the projected criteria for involution were obtained (that is a zero dimensional symbol in that case). This has the advantage that geometric involutive form calculations whose complexity implicitly depends on the total number of complex solutions are avoided until later, when such complex solutions have been discarded as a result of new generators being found in the kernel of the moment matrix.

## Bibliography

- [1] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Math.* Springer-Verlag, 2 edition, 2006. 25, 28, 55, 56
- [2] Daniel J Bates, Jonathan D Hauenstein, Andrew J Sommese, and Charles W Wampler. *Numerically solving polynomial systems with Bertini*, volume 25. SIAM, 2013. 4, 25
- [3] Gian Mario Besana, Sandra Di Rocco, Jonathan D Hauenstein, Andrew J Sommese, and Charles W Wampler. Cell decomposition of almost smooth real algebraic surfaces. *Numerical Algorithms*, 63(4):645–678, 2013. 26
- [4] G. Blekherman, P.A. Parrilo, and R.R. Thomas, editors. *Semidefinite optimization and convex algebraic geometry*, volume 13 of *MOS-SIAM Series on Optimization*. Society

for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA, 2013. 25, 55, 56, 68, 94

- [5] J. Bochnak, M. Coste, and M.F. Roy. *Real Algebraic Geometry*. Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge A Series of Modern Surveys in Mathematics. Springer, 1998. 28, 63
- [6] J. Bonasia, F. Lemaire, G.J. Reid, and L. Zhi. Determination of approximate symmetries of differential equations. *Group Theory and Numerical Analysis*, 39:249, 2005. 24, 31, 33, 35, 59, 61, 112
- [7] Changbo Chen, James H Davenport, John P May, Marc Moreno Maza, Bican Xia, and Rong Xiao. Triangular decomposition of semi-algebraic systems. In *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, pages 187–194. ACM, 2010. 25, 46, 129
- [8] Changbo Chen, Marc Moreno Maza, Bican Xia, and Lu Yang. Computing cylindrical algebraic decomposition via triangular decomposition. In *Proceedings of the 2009 international symposium on Symbolic and algebraic computation*, pages 95–102. ACM, 2009. 25, 46, 129
- [9] George E Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 20–23, 1975*, pages 134–183. Springer, 1975. 25
- [10] RE Curto and LA Fialkow. Solution of the truncated complex moment problem for flat data-introduction. *Memoirs of the American Mathematical Society*, 119(568):1, 1996. 13, 15, 16, 17, 25, 108, 109, 110
- [11] James H Davenport and Joos Heintz. Real quantifier elimination is doubly exponential. *Journal of Symbolic Computation*, 5(1):29–35, 1988. 25
- [12] Jean-Charles Faugere. A new efficient algorithm for computing gröbner bases. *Journal of pure and applied algebra*, 139(1):61–88, 1999. 46, 129

- [13] V.P. Gerdt and Y.A. Blinkov. Involutive bases of polynomial ideals. *Mathematics and Computers in Simulation*, 45(5):519–541, 1998. 7, 24, 29, 59, 112
- [14] Patrizia Gianni, Barry Trager, and Gail Zacharias. Grbner bases and primary decomposition of polynomial ideals. *Journal of Symbolic Computation*, 6(23):149 – 167, 1988. 27
- [15] D Yu Grigor’ev and NN Vorobjov Jr. Counting connected components of a semialgebraic set in subexponential time. *Computational Complexity*, 2(2):133–186, 1992. 26
- [16] Jonathan D Hauenstein. Numerically computing real points on algebraic sets. *Acta applicandae mathematicae*, 125(1):105–119, 2013. 5, 26, 83, 85, 121
- [17] Hoon Hong. *Improvements in Cad-based Quantifier Elimination*. PhD thesis, The Ohio State University, 1990. AAI9111716. 25
- [18] M. Kuranishi. On e. cartan’s prolongation theorem of exterior differential systems. *American Journal of Mathematics*, pages 1–47, 1957. 24, 29, 33, 59, 112
- [19] J.B. Lasserre, M. Laurent, and P. Rostalski. A prolongation–projection algorithm for computing the finite real variety of an ideal. *Theoretical Computer Science*, 410(27):2685–2700, 2009. 1, 8, 25, 47, 53, 54, 55, 64, 84, 93, 94, 120
- [20] M. Laurent and P. Rostalski. The approach of moments for polynomial equations. In Miguel F. Anjos and Jean B. Lasserre, editors, *Handbook on semidefinite, conic and polynomial optimization*, International Series in Operations Research & Management Science, 166, pages 25–60. Springer, New York, 2012. 10, 13, 14, 15, 16, 17, 36, 38, 40, 46, 47, 58, 95, 96, 128, 129
- [21] Ye Lu. *Finding all real solutions of polynomial systems*. PhD thesis, University of Notre Dame, 2006. 26
- [22] Y. Ma and L. Zhi. Computing real solutions of polynomial systems via low-rank moment matrix completion. In *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation*, pages 249–256. ACM, 2012. 26, 83

- [23] F.S. Macaulay and P. Roberts. *The algebraic theory of modular systems*. University press Cambridge, 1916. 29, 57
- [24] H.M. Möller and T. Sauer. H-bases for polynomial interpolation and system solving. *Advances in Computational Mathematics*, 12(4):335–362, 2000. 29, 46, 57, 83
- [25] B. Mourrain. Isolated points, duality and residues. *Journal of Pure and Applied Algebra*, 117:469–493, 1997. 29, 46, 57
- [26] B. Mourrain. A new criterion for normal form algorithms. In *Applied algebra, algebraic algorithms and error-correcting codes*, pages 430–442. Springer, 1999. 29, 46, 57
- [27] B. Mourrain and P. Trebuchet. Solving projective complete intersection faster. In *Proceedings of the 2000 international symposium on Symbolic and algebraic computation*, pages 234–241. ACM, 2000. 46
- [28] B. Mourrain and P. Trébuchet. Algebraic methods for numerical solving. In *Proc. of the 3rd International Workshop on Symbolic and Numeric Algorithms for Scientific Computing*, volume 1, pages 42–57, 2002. 46
- [29] Jean-François Pommaret. *Systems of partial differential equations and Lie pseudogroups*, volume 14. CRC Press, 1978. 29, 33
- [30] G.J. Reid, C. Smith, and J. Verschelde. Geometric completion of differential systems using numeric-symbolic continuation. *ACM SIGSAM Bulletin*, 36(2):1–17, 2002. 33
- [31] G.J. Reid, J. Tang, and L. Zhi. A complete symbolic-numeric linear method for camera pose determination. In *Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, pages 215–223. ACM, 2003. 24, 31, 33, 59, 112
- [32] G.J. Reid and L. Zhi. Solving polynomial systems via symbolic-numeric reduction to geometric involutive form. *Journal of Symbolic Computation*, 44(3):280–291, 2009. 24, 59, 112
- [33] Jean-Jacques Risler. Une caractérisation des idéaux des variétés algébriques réelles. *CR Acad. Sci. Paris*, 271:1171–1173, 1970. 28, 63

- [34] Fabrice Rouillier, M-F Roy, and M Safey El Din. Finding at least one point in each connected component of a real algebraic set defined by a single equation. *Journal of Complexity*, 16(4):716–750, 2000. 26
- [35] R. Scott, G.J. Reid, W. Wu, and L. Zhi. Geometric involutive bases and applications to approximate commutative algebra. In Lorenzo Robbiano and John Abbott, editors, *Approximate Commutative Algebra*, pages 99–124. Springer, 2010. 24, 37, 43, 55, 59, 60, 95, 112
- [36] Abraham Seidenberg. A new decision method for elementary algebra. *Annals of Mathematics*, pages 365–374, 1954. 25
- [37] Werner M Seiler. *Involution: The formal theory of differential equations and its applications in computer algebra*, volume 24 of *Algorithms and Computation in Mathematics*. Springer, 2010. 29, 112
- [38] A.J. Sommese and C.W. Wampler. *The Numerical solution of systems of polynomials arising in engineering and science*, volume 99. World Scientific, 2005. 1, 4, 25, 53, 93
- [39] Hans J. Stetter. *Numerical polynomial algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2004. 29, 57, 84, 85, 120, 129
- [40] Alfred Tarski. A decision method for elementary algebra and geometry. In *Quantifier elimination and cylindrical algebraic decomposition*, pages 24–84. Springer, 1998. 25
- [41] A.D. Wittkopf and G.J. Reid. Fast differential elimination in c: The cdiffelim environment. *Computer Physics Communications*, 139(2):192–217, 2001. 24, 31, 33, 59, 112, 128
- [42] W. Wu and G.J. Reid. Finding points on real solution components and applications to differential polynomial systems. In *Proceedings of the 38th international symposium on International symposium on symbolic and algebraic computation*, pages 339–346. ACM, 2013. 5, 26, 83, 85, 121

- [43] X. Wu and L. Zhi. Determining singular solutions of polynomial systems via symbolic–numeric reduction to geometric involutive forms. *Journal of Symbolic Computation*, 47(3):227–238, 2012. 24, 59, 112
- [44] Z. Yang, L. Zhi, and Y. Zhu. Verified error bounds for real solutions of positive-dimensional polynomial systems. In *Proceedings of the 38th international symposium on International symposium on symbolic and algebraic computation*, pages 371–378. ACM, 2013. 26



## Chapter 3

# Semidefinite Programming and facial reduction for Systems of Polynomial Equations

### 3.1 Introduction

The breakthrough work of Lasserre and collaborators [32, 49] shows that the *real radical ideal*, *RRI*, of a real polynomial system with finitely many solutions can be determined by a combination of a *semidefinite programming*, *SDP*, feasibility problem and the *geometric involutive form*, *GIF*. This RRI is generated by a system of real polynomials having only real roots that are free of multiplicities. Global numerical solvers, such as homotopy continuation solvers typically compute all real roots by first computing all complex (including real) roots. And if the roots have multiplicity, then elaborate strategies are needed to avoid difficulties that arise as the paths from the homotopy solvers approach these (singular Jacobian) roots [48]. Furthermore, random polynomial systems of  $k$  real polynomials of degree  $d$  in  $n$  variables can have  $d^n$  roots, and if the coefficients follow a certain probability distribution have only  $d^{n/2}$  real roots on average, see [23] and the references therein. Therefore, consideration of only the real roots simplifies the problem. A conjectured extension of such methods to positive dimensional polynomial systems has been given recently by Ma, Wang and Zhi [37, 36]. These extensions

depend on the *method of moments* within a SDP formulation.

Our SDP feasibility formulation is a moment problem equivalent to finding  $X$  for a linear system of the following type (also Problem 3.1.1 below)

$$\mathcal{A}X = b, \quad X \in \mathcal{S}_+^k, \quad (3.1)$$

where  $\mathcal{S}_+^k$  denotes the convex cone of  $k \times k$  real symmetric positive semi-definite matrices, and  $\mathcal{A} : \mathcal{S}_+^k \rightarrow \mathbb{R}^m$  is a linear transformation. The standard regularity assumption for (3.1) is the *Slater constraint qualification* or strict feasibility assumption:

$$\text{there exists } \hat{X} \text{ with } \mathcal{A}\hat{X} = b, \quad \hat{X} \in \text{int } \mathcal{S}_+^k. \quad (3.2)$$

We let  $X \geq 0, > 0$  denote  $X \in \mathcal{S}_+^k, \in \text{int } \mathcal{S}_+^k$ , respectively. It is well known that the Slater condition for SDP holds generically, e.g., [21]. Surprisingly, many SDP problems arising from particular applications, and in particular our polynomial system applications, are marginally infeasible, i.e., fail to satisfy strict feasibility. This means that the feasible set lies within the boundary of the cone, and even the slightest perturbation of the data can make the problem infeasible. This creates difficulties with the optimality and duality conditions as well as with numerical algorithms. To help regularize such SDP problems so that *strong duality* holds, facial reduction was introduced in 1982 by Borwein and Wolkowicz [13, 14]. However it was only much later that the power of facial reduction was exhibited in many applications, e.g., [56, 53, 1]. Developing algorithmic implementations of facial reduction that work for large classes of SDP problems and the connections with perturbation and convergence analysis has recently been achieved in e.g., [30, 19, 16, 20].

A polynomial system of maximum degree  $d$  equations in  $n$  variables can be viewed as the equation  $Cx = 0$ , a function of its monomials [32, 49]. Here  $x$  is a vector of the  $N(n, d) = \frac{(d+n)!}{d!n!} = \binom{d+n}{d}$  monomials up to the degree  $d$  of the polynomial system. This equation yields part of the system of linear constraints in the SDP formulation of polynomial systems. The convex cone for polynomials are semi-definite moment matrices encoding the real solutions of the polynomial equations and certain generalized Hankel-Macaulay structure possessed by the polynomial systems. Remarkable advances have been recently made in this area

[32, 49, 9] which is an intersection between optimization and algebraic geometry. In this article we establish a framework for using facial reduction for such systems and then solving the systems using the regularized smaller SDP. We note that familiar methods for linear systems of equations when  $d = 1$  are *Gaussian elimination*, *GE*, for exact solutions and *singular value decompositions*, *SVD*, for least squares solutions. For polynomial systems, the corresponding method in the exact case uses *Gröbner Bases* [4]. A major difference for Gröbner Bases to the case  $d = 1$  is that generalized row operations involving multiplication by monomials and not just scalars is permitted. The operation of multiplying a polynomial by such a monomial raises its degree and is called prolongation. Eliminating between prolonged equations, is called projection. In the approximate case, as in our paper, we use *geometric involutive bases* [47] which use the SVD.

In particular a polynomial system can possess constraints resulting from this process that are *higher* than the degree of the system. So in this paper, as in [32, 49] and in Ma, Wang and Zhi [37, 36], higher degree systems can result. This continual extension of the underlying space is a significant practical and theoretical challenge in algorithm development.

The RRI of our system  $P$  is the set of all polynomials with the same zero set as  $P$ . To give the reader an informal introduction to RRIs and their interpretation, consider the simple case of *univariate polynomials* with real coefficients,  $n = 1$ . In this case, the factors of the coefficients are either complex or real. The RRI discards the complex factors and also the multiplicities from the polynomial, to obtain a new polynomial. This reduced polynomial is the generating polynomial for the RRI of the original polynomial, and has the same real roots, no multiplicities and no complex roots.

Combining SDP methods and applying them to a polynomial system  $P$  with coefficient matrix  $C(P)$  and associated moment matrix  $M(u) \in \mathbb{R}^{N(n,d) \times N(n,d)}$  yields the following problem central to our paper:

**Problem 3.1.1 (Moment Matrix Feasibility Problem)** Find  $u \in \mathbb{R}^{N(n,2d)}$  where  $N(n,d) = \binom{d+n}{d}$  so that

$$C(P)M(u) = 0, \quad M_{11}(u) = 1, \quad M(u) \geq 0.$$

Also see Problem 3.5.1 in Section 3.5.

We continue in Section 3.2 with material on real polynomial systems, their RRI's and the coefficient matrix representations. In Section 3.3 we give a condensed and more formal description of geometric involutive bases and the related algorithms. In Section 3.4 we combine the moment matrix and geometric involutive form algorithms to yield our fundamental Algorithm 3.4.1 for polynomial systems. In particular Algorithm 3.4.1 proceeds by putting the polynomials into GIF using Algorithm 3.3.1; we then solve the related moment matrix problem using Algorithm 3.2.1. These two steps are iterated until satisfaction of the Rank-Dim-Involutive Stopping Criterion 3.10.

In Section 3.5 we describe the facial reduction and projection methods for finding feasible solutions for the moment matrix feasibility problem 3.1.1. We also describe the *Douglas-Rachford (DR)* projection/reflection method that we use. We also present our implementation of facial reduction. Section 3.6 gives the numerical experiments. Our concluding remarks are in Section 3.7.

## 3.2 Real radical ideals and moment matrices

We now present some material on real polynomial systems, their RRI's and the coefficient matrix representation needed for our paper. For background and references to real algebraic geometry see e.g., [4, 9, 49, 2].

### 3.2.1 Real polynomial systems

We consider a (finite) system of  $m$  polynomials in  $n$  variables

$$P := \{p_1, \dots, p_m\} \subset \mathbb{R}[x_1, \dots, x_n] =: \mathbb{R}[x],$$

where  $\mathbb{R}[x]$  is the set of all polynomials with real coefficients in the  $n$  variables  $x = (x_1, x_2, \dots, x_n)^T$ .

We let  $d = \deg(P)$  denote the *degree of the polynomial system*, i.e., the maximum of the degrees of the polynomials  $p_j$  in  $P$ . The solution set or variety of  $P$  is

$$V_{\mathbb{K}}(p_1, \dots, p_m) = \{x \in \mathbb{K}^n : p_j(x) = 0, \forall 1 \leq j \leq m\}. \quad (3.3)$$

This is the *real variety* of  $P$  if  $\mathbb{K} = \mathbb{R}$  and the *complex variety* of  $P$  if  $\mathbb{K} = \mathbb{C}$ . The real ideal generated by  $P = \{p_1, \dots, p_m\} \subset \mathbb{R}[x]$  is:

$$\langle P \rangle_{\mathbb{R}} = \langle p_1, \dots, p_m \rangle_{\mathbb{R}} = \{f_1 p_1 + \dots + f_m p_m : f_j \in \mathbb{R}[x], \forall 1 \leq j \leq m\}. \quad (3.4)$$

We denote a *monomial* by  $x^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ , where  $\alpha \in \mathbb{N}^n$ ,  $\mathbb{N}$  is the set of nonnegative integers. The *degree of the monomial* is  $|\alpha| := \|\alpha\|_1 = \alpha_1 + \dots + \alpha_n$ . It is clear that the degree of each monomial satisfies  $|\alpha| \leq d$ , the degree of the polynomial. Throughout this paper we use *graded reverse lexicographic order*, *grevlex*, to order the set of monomials.<sup>1</sup>

We can rewrite the system of  $m$  polynomials,  $P$ , as

$$P = \left\{ \sum_{|\alpha| \leq d} a_{k,\alpha} x^\alpha : k = 1, \dots, m \right\}. \quad (3.5)$$

This order respects the *Cartan class of variables*, which is important in our numerical determination of the geometric features of the polynomial systems such as those in Definition 3.3.3 below.

**Definition 3.2.1 (Coefficient matrix of  $P$ ,  $C(P)$ )** Let  $x^{(\leq d)} = (x^\alpha)$  be the column vector of monomials  $x^\alpha$  with  $0 \leq |\alpha| \leq d$  ordered as in *grevlex* above. Suppose that the coefficients  $a_{k,\alpha}$  in (3.5) are similarly ordered. Then define the coefficient matrix of  $P$  by  $C(P) = (a_{k,\alpha})$ .

The following lemma follows immediately.

**Lemma 3.2.1** With  $C(P)$ ,  $x^{(\leq d)}$  defined in Definition 3.2.1, we have

$$P = C(P)x^{(\leq d)}, \quad (3.6)$$

with  $C(P) \in \mathbb{R}^{m \times N(n,d)}$  and  $N(n,d) := \binom{d+n}{d}$  is the number of monomials in  $x^{(\leq d)}$ .

The well known presentation of polynomial systems as linear functions of their monomials along with the related coefficient matrix and its kernel and rowspace has been exploited in [50, 41, 42, 40] and in the historical work by Macaulay [39]. For an introductory example see [44].

---

<sup>1</sup>This is often called *grevlex* in the literature. It compares the total degree first and then compares exponents of the last indeterminate but while reversing the outcome so that the monomial with smaller exponent is larger in the ordering.

### 3.2.2 Moment matrices

Moment matrices  $M(\mu)$  arise as a means of representing real polynomial systems. We outline the procedure for finding  $M(\mu)$  in Algorithm 3.2.1. For theoretical background the reader is directed to e.g., [2, 33].

A moment matrix is an infinite real symmetric matrix  $M = (M_{\alpha,\beta})$  with indices corresponding to the indices of the monomials  $\alpha, \beta \in \mathbb{N}^n$ . Here  $\alpha$  is the index for rows and  $\beta$  is the index for columns. Without loss of generality, we assume that  $M_{0,0} = 1$ . The matrix arises from considering the product of monomials  $x^\alpha x^\beta = x^{\alpha+\beta}$  and then the correspondence  $u_\alpha \leftrightarrow x^\alpha$  extends to the formal correspondence  $x^\alpha x^\beta \leftrightarrow u_{\alpha+\beta}$ .

**Definition 3.2.2 (Moment matrix)** *Let  $u = \{u_\alpha : \alpha \in \mathbb{N}^n, |\alpha| \leq d\} \in \mathbb{R}^{N(n,d)}$  be a vector of indeterminates where the entries are indexed corresponding to the exponent vectors of the monomials in  $n$  variables of degree at most  $d$ . The degree  $d$  moment matrix of  $u$  is a  $N(n,d) \times N(n,d)$  symmetric matrix with rows and columns corresponding to monomials in  $n$  variables of degree at most  $d$ , and defined as*

$$M(u) = \left[ u_{\alpha+\beta} \right]_{|\alpha|, |\beta| \leq d}.$$

Given a multivariate polynomial system  $P \subset \mathbb{R}[x]$  with  $d = \deg(P)$  we let  $M$  denote the truncated moment matrix.

**Lemma 3.2.2** *The truncated moment matrix  $M \in \mathcal{S}_+^{N(n,d)}$ . The linear constraints imposed by  $P$  from (3.6) are  $C(P)M = 0$ , where  $C(P)$  is the coefficient matrix function given in Definition 3.2.1. ■*

**Example 3.2.1 (Moment matrix for univariate example  $x = (x_1)$ )** *The moment matrix in the univariate ( $n = 1$ ) case is the infinite matrix whose  $(\alpha, \beta)$  entry is  $u_{\alpha+\beta}$  and  $\alpha, \beta \in \mathbb{N}$  given by:*

$$M(u) = \begin{bmatrix} u_0 & u_1 & u_2 & u_3 & u_4 & \cdots \\ u_1 & u_2 & u_3 & u_4 & u_5 & \cdots \\ u_2 & u_3 & u_4 & u_5 & u_6 & \cdots \\ u_3 & u_4 & u_5 & u_6 & u_7 & \cdots \\ u_4 & u_5 & u_6 & u_7 & u_8 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad u_0 = 1. \quad (3.7)$$

Note that (3.7) is a Hankel matrix. Let us associate  $u_\alpha \leftrightarrow x^\alpha$ . Then we recover the polynomial equation using the coefficient matrix as  $C(P)x^{(\leq d)}$ . This implies that in terms of the moment matrix, we get  $C(P)M(u) = 0$ .

---

**Algorithm 3.2.1: M - Moment Matrix**


---

- 1 Input(  $P \subset \mathbb{R}[x_1, \dots, x_n]$ . Set  $d := \deg(P)$ );
  - 2 Use an SDP method to find a maximum rank moment matrix  $M(\mu^*)$  with the additional coefficient constraint  $C(P) M(u^*) = 0$ ;
  - 3 Output(  $M(u^*) \geq 0$ , the maximum rank moment matrix)
- 

### 3.3 Geometric involutive bases

In this section we introduce the basic objects for geometric involutive bases. Algorithm 3.3.1 finds the GIF. For more details and examples see [44, 11].

Involutivity originates in the geometry of differential equations. See Kuranishi [31] for a famous proof of termination of Cartan's prolongation algorithm for nonlinear partial differential equations. A by-product of these methods has been their implementation for linear homogeneous partial differential equations with constant coefficients, and consequently for polynomial algebraic systems. See [26] for applications and symbolic algorithms for polynomial systems. The symbolic-numeric version of a *geometric involutive form*, *GIF*, was first described and implemented in Wittkopf and Reid [51]. It was applied to approximate symmetries of differential equations in [11] and to polynomial solving in [45, 43, 47]. See [55] where it is applied to the deflation of multiplicities in multivariate polynomial solving.

**Definition 3.3.1** *Let  $P$  be a finite subset of  $\mathbb{R}[x]$  of degree  $d$ . The  $k$ -th prolongation of system  $P$  is*

$$\widehat{D}^k(P) := \{x^\alpha p : 0 \leq \deg(x^\alpha p) \leq d + k, \alpha \in \mathbb{N}^n, p \in P\}.$$

For example  $\widehat{\mathbf{D}}^1(P)$  for  $P = \{x^2 - x - 1, xy - y - 1\}$  consists of  $P$  together with the 4 polynomials in

$$\begin{aligned} x(x^2 - x - 1) &= x^3 - x^2 - x \\ x(xy - y - 1) &= x^2y - xy - x \\ y(x^2 - x - 1) &= x^2y - xy - y \\ y(xy - y - 1) &= xy^2 - y^2 - y. \end{aligned} \tag{3.8}$$

We can *project* by eliminating higher degree monomials in favour of lower degree ones. In the prolonged system we can project the system from degree 3 to degree 2 by eliminating the highest degree term  $x^2y$  as well as  $xy$  that occurs in the second and third equations of (3.8) to obtain the new projected equation  $y - x = 0$ .

**Definition 3.3.2** *Given a subspace  $V$  of  $J^d := \mathbb{R}^{N(n,d)}$  and  $m \leq d$ , define  $\pi^m(V)$  as the vectors of  $V$  with the components of degree  $\geq d - m$  discarded. Given  $P \subset \mathbb{R}[x]$  of degree  $d$  define  $\pi^m(P) := \pi^m \ker C(P)$ . The  $k$ -th prolongation of the kernel is  $\widehat{\mathbf{D}}^k(P) := \ker C(\widehat{\mathbf{D}}^k P)$ .*

See for example [47] and the references in [44] for the stable numerical implementations of this paper's operations using SVD methods. In Remark 3.5 of [44] we discuss how prolongation and projection can equivalently be computed in the kernel or rowspace, and how polynomial generators can always be extracted. Underlying this is a 1 – 1 correspondence between the relevant vector spaces (not elements).

**Definition 3.3.3 (Symbol, class and Cartan involution test)** *Suppose  $P \subset \mathbb{R}[x]$  of degree  $d$ . The symbol matrix  $\mathcal{S}(P)$  of  $P$  is the submatrix of  $C(P)$  corresponding to its degree  $d$  monomials. Then the class of a monomial  $x^\alpha$  is the least  $j$  such that  $\alpha_j \neq 0$ .*

Suppose that the columns of  $\mathcal{S}(P)$  are sorted in descending order by class and that it is reduced to Gauss echelon form. For  $k = 1, 2, \dots, n$  define the quantities  $\beta_d^{(k)}$  as the number of pivots in this reduced matrix of class  $k$ . In a generic system of coordinates the symbol is involutive if

$$\sum_{k=1}^{k=n} k\beta_d^{(k)} = \text{rank } \mathcal{S}(\widehat{\mathbf{D}}P) \tag{3.9}$$



Suppose  $Q \subset \mathbb{R}[x]$  has degree  $d'$  and a basis for  $\ker C(Q)$  is given by the rows of the matrix  $B$ . To extract the  $\beta_q^{(k)}$  in (3.9) at projected degree  $d \leq d'$  we first numerically project  $\ker C(Q)$  onto the subspace  $J^d$  by deleting the coordinates in  $B$  of degree  $> d$  to give a spanning set  $\tilde{B}$  for  $\pi^{d'-d}Q$ . Then delete the columns in  $\tilde{B}$  corresponding to variables of degree  $< d$  to obtain a matrix  $A_d$  corresponding to the orthogonal complement of the degree  $d$  symbol. Let  $A_d^{(k)}$  be the submatrix of  $\tilde{B}$  with columns corresponding to variables of class  $\leq k$ . In generic coordinates for  $k = 1 \dots n$ :

$$\beta_d^{(k)} = \binom{n + d - k - 1}{d - 1} - (\text{rank } A_d^{(k-1)} - \text{rank } A_d^{(k)}) .$$

Then the SVD can approximate the ranks in this equation for carrying out the Cartan Test (3.9).

**Definition 3.3.4 (Involutive System)** *A system of polynomials  $P \subset \mathbb{R}[x]$  is involutive if  $\dim \pi \mathbf{D}P = \dim P$  and the symbol of  $P$  is involutive.*

**Definition 3.3.5** *Let  $P \in \mathbb{R}[x]$  with  $d = \deg P$  and  $k, m$  be integers with  $k \geq 0$  and  $0 \leq m \leq k + d$ . Then  $\pi^m \mathbf{D}^k P$  is projectively involutive if  $\dim \pi^m \mathbf{D}^k P = \dim \pi^{m+1} \mathbf{D}^{k+1} P$  and the symbol of  $\pi^m \mathbf{D}^k P$  is involutive.*

In [11] it is proved that a system is projectively involutive if and only if it is involutive. In Algorithm 3.3.1 we seek the smallest  $k$  such that there exists an  $m$  with  $\pi^m \mathbf{D}^k P$  approximately involutive, and generates the same ideal as the input system. We choose the system corresponding to the largest such  $m \leq k$  if there are several such values for the given  $k$ .

---

**Algorithm 3.3.1:** GIF: Geometric involutive form

---

```

1  Input(  $P \subset \mathbb{R}[x_1, \dots, x_n]$ ; tolerance  $\epsilon$ .);
2  Set  $k := 0$ ,  $d := \deg(P)$  and  $B$  for  $\ker C(P)$ ,  $J = \{\}$  ;
3  repeat
4      Compute  $\mathbf{D}^k(P)$ ; initialize set of involutive systems  $I := \{\}$  ;
5      for  $j$  from 0 to  $(d + k)$  do
6          Compute  $R := \pi^j \mathbf{D}^k(P)$ ;
7          if  $R$  involutive then
8               $I := I \cup \{R\}$ 
9          end if
10     end for
11     Select all  $\bar{R}$  from  $I$ :  $\mathbf{D}^{d+k-\bar{d}} \bar{R} \subseteq \mathbf{D}^k(P)$  where  $\bar{d} = \deg(\bar{R})$  ;
12     Place the selected involutive  $\bar{R}$  from  $I$  in the set  $J$  ;
13      $k := k + 1$ 
14 until  $J \neq \emptyset$ ;
15 Output( Return  $R = \text{GIF}(P)$  the polynomial generators of the involutive system in  $J$  of
lowest degree.)
```

---

The degree of the geometric involutive basis in our method can be lower than that given in [37, 36] since Algorithm 3.3.1 updates the generators with projections. However, in the absence of a proof of determination of the real radical, we conclude that the larger moment matrices of [37] can capture new members of the real radical in situations where our method has already terminated.

Additional discussion and examples are given in the long version of our work [44].

### 3.4 Combining the moment matrix and geometric involutive form algorithms

The complete method that combines the moment matrix and geometric involution techniques is given in Algorithm 3.4.1.

Recall that  $M = M(u) = (M_{\alpha,\beta})$  denotes the moment matrix indexed by  $\alpha, \beta$  for rows and columns, respectively. And,  $d = \deg(P)$ ,  $M \in \mathcal{S}^{N(n,d)}$ , and the linear constraints imposed by our system of polynomials  $P \subset \mathbb{R}[x]$  are given using the coefficient matrix  $C(P)M = 0$ . We let  $\langle P \rangle_{\mathbb{R}}$  denote the *associated polynomial ideal* and let

$$\sqrt[\mathbb{R}]{\langle P \rangle_{\mathbb{R}}} = \{f \in \mathbb{R}[x] : f^{2m} + \sum_{j=1}^s q_j^2 \in \langle P \rangle_{\mathbb{R}}, q_j \in \mathbb{R}[x], m \in \mathbb{N}_+\}$$

denote the RRI generated by polynomials  $P$  over  $\mathbb{R}$ . A fundamental result [10] (originally proved in [46]) called the *Real Nullstellensatz* is

$$\sqrt[\mathbb{R}]{\langle P \rangle_{\mathbb{R}}} = \{f(x) \in \mathbb{R}[x] : f(x) = 0, \forall x \in V_{\mathbb{R}}(P)\}.$$

Algorithm 3.4.1 proceeds by putting the polynomials into GIF using Algorithm 3.3.1; we then solve the related moment matrix problem using Algorithm 3.2.1. These two steps are iterated until satisfaction of the Rank-Dim-Involutive Stopping Criterion 3.10, that is  $r = d$ . If the ideal generated by the output system is zero dimensional then the output is a GIF for the real radical which is proved later in Chapter 4 by Theorem 4.7.5 and Theorem 4.7.6. If the input system is positive dimensional, then the output is a GIF for an intermediate idea between the input ideal and the real radical.

**Algorithm 3.4.1:** GIF – SDP Method

---

```

1 Input(  $P = \{p_1, \dots, p_k\} \subset \mathbb{R}[x_1, \dots, x_n]$ );
2 Set  $P_0 := P, \quad j := 0$ ;
3 repeat
4    $d := \dim \ker \text{GIF}(P_j), \quad P_{j+1} := \text{GIF}(P_j)$ ;
5   Find  $u^* \in \mathbb{R}^{N(n, 2d)}$ :  $M(u^*) \geq 0, C(P_{j+1})M(u^*) = 0$  (Described in Algorithm 3.2.1);
6    $r := \text{rank}(M(u^*)), \quad P_{j+2} := \text{gen}(\ker M(u^*))$ ;
7    $j := j + 2$ 
8 until  $r = d$ ;
9 Output( $P_{j+1} \subset \mathbb{R}[x_1, \dots, x_n]$ ;  $P_{j+1}$  is in geometric involutive form ;
 $\sqrt[n]{\langle P \rangle_{\mathbb{R}}} \supseteq \langle P_{j+1} \rangle_{\mathbb{R}} \supseteq \langle P \rangle_{\mathbb{R}.}$ )

```

---

The Algorithms 3.2.1, 3.3.1, and 3.4.2 are *subroutines* for our principal Algorithm 3.4.1.

**Algorithm 3.4.2:** gen

---

```

1 Input(  $\ker M(u^*)$  where  $M(u^*)$  is the optimal max-rank moment matrix.);
2 Output(Polynomial generators corresponding to  $\ker M(u^*)$ )

```

---

**Rank-Dim-Involutive Stopping Criterion** The natural termination criterion used in Algorithm 3.4.1 is that:

$$\dim \ker \text{GIF}(P_j) = d = r = \text{rank}(M(u^*)) \text{ and } P_j \text{ involutive,} \quad (3.10)$$

where  $u^*$  corresponds to the optimal moment matrix  $M(u^*)$ . From results in [32],  $\langle \text{gen}(\ker M(P_{j+1})) \rangle$  is a sequence of ideals contained in  $\sqrt[n]{\langle P \rangle}$ . We get an ascending chain of ideals in a Noetherian ring  $\mathbb{R}[x_1, \dots, x_n]$ . Hence, together with the finiteness of the Cartan-Kuranishi geometric involutive form algorithm, Algorithm 3.4.1 terminates in a finite number of steps.

### 3.5 Facial reduction and projection methods

In this section we describe the facial reduction and projection methods for finding feasible solutions for the moment matrix feasibility problem. Our moment problem is given in Problem 3.5.1, where  $M(u)$  implicitly denotes the moment matrix constraints, i.e., the intersection of the space of generalized Hankel matrices with the semidefinite cone.

**Problem 3.5.1 (Moment Matrix Feasibility Problem)** *Let  $C = C(P)$  be a given  $N(n, d) \times m$  (coefficient) matrix of full column rank. Find  $u \in \mathbb{R}^{N(n, 2d)}$  so that*

$$C^T M(u) = 0, \quad M(u)_{11} = 1, \quad M(u) \geq 0.$$

#### 3.5.1 Representations for linear constraints for moment problems

An important initial step for our methods is building an efficient (onto) matrix representation for the linear constraints on the moment matrices resulting from the polynomial systems. Recall that we introduced moment matrices informally by a simple example in Section 3.2.2; see also Definition 3.2.2. Let  $u_\alpha := u_{(\alpha_1, \dots, \alpha_n)}$  where  $\alpha \in \mathbb{N}^n$  and the degree of  $u_\alpha$  is  $|\alpha| = \alpha_1 + \dots + \alpha_n$ . Let  $(u_{(\alpha \leq d)})$  be an array of the  $u_\alpha$ 's with  $0 \leq |\alpha| \leq d$  and sorted in grevlex order as described above.

Consider a truncated moment matrix  $M(u) = (u_{\alpha+\beta})_{\alpha, \beta \in \mathbb{N}^n, |\alpha|, |\beta| \leq d}$ . The generalized truncated moment matrix can be represented as follows, where

$$\langle f_i(u), f_j(u) \rangle_* = u(i) + u(j).$$

We assume the length of  $\langle u_{(\alpha \leq d)} \rangle$  is  $k + 1$ . (We provide a formula for  $k$  in Algorithm 3.5.1 below.)

$$M(u) = \begin{bmatrix} \langle f_0(u), f_0(u) \rangle_* & \langle f_0(u), f_1(u) \rangle_* & \langle f_0(u), f_2(u) \rangle_* & \dots & \langle f_0(u), f_k(u) \rangle_* \\ \langle f_1(u), f_0(u) \rangle_* & \langle f_1(u), f_1(u) \rangle_* & \langle f_1(u), f_2(u) \rangle_* & \dots & \langle f_1(u), f_k(u) \rangle_* \\ \langle f_2(u), f_0(u) \rangle_* & \langle f_2(u), f_1(u) \rangle_* & \langle f_2(u), f_2(u) \rangle_* & \dots & \langle f_2(u), f_k(u) \rangle_* \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \langle f_k(u), f_0(u) \rangle_* & \langle f_k(u), f_1(u) \rangle_* & \langle f_k(u), f_2(u) \rangle_* & \dots & \langle f_k(u), f_k(u) \rangle_* \end{bmatrix}$$

In the univariate case the moment matrices have Hankel structure as shown in (3.7). In Table 3.1 we display a truncated bivariate moment matrix partitioned into block submatrices

having the same degree. Notice that the matrix in Table 3.1 is not Hankel. However each

$$M(u) = \begin{bmatrix} u_{00} & u_{10} & u_{01} & u_{20} & u_{11} & u_{02} & u_{30} & u_{21} & u_{12} & u_{03} \\ u_{10} & u_{20} & u_{11} & u_{30} & u_{21} & u_{12} & u_{40} & u_{31} & u_{22} & u_{13} \\ u_{01} & u_{11} & u_{02} & u_{21} & u_{12} & u_{03} & u_{31} & u_{22} & u_{13} & u_{04} \\ u_{20} & u_{30} & u_{21} & u_{40} & u_{31} & u_{22} & u_{50} & u_{41} & u_{32} & u_{23} \\ u_{11} & u_{21} & u_{12} & u_{31} & u_{22} & u_{13} & u_{41} & u_{32} & u_{23} & u_{14} \\ u_{02} & u_{12} & u_{03} & u_{22} & u_{13} & u_{04} & u_{32} & u_{23} & u_{14} & u_{05} \\ u_{30} & u_{40} & u_{31} & u_{50} & u_{41} & u_{32} & u_{60} & u_{51} & u_{42} & u_{33} \\ u_{21} & u_{31} & u_{22} & u_{41} & u_{32} & u_{23} & u_{51} & u_{42} & u_{33} & u_{24} \\ u_{12} & u_{22} & u_{13} & u_{32} & u_{23} & u_{14} & u_{42} & u_{33} & u_{24} & u_{15} \\ u_{03} & u_{13} & u_{04} & u_{23} & u_{14} & u_{05} & u_{33} & u_{24} & u_{15} & u_{06} \end{bmatrix}$$

Table 3.1: block partitioned bivariate moment matrix; submatrices have same degree

of its block matrices is rectangular Hankel; though even this feature is lost for multivariate moment matrices in more than two variables. As mentioned above, without loss of generality we assume that  $u_{00} = 1$ .

Besides being a symmetric matrix, the moment matrix also has other linear constraints among its entries. One can easily see these constraints in the truncated univariate matrix (3.7) and bivariate matrix in Table 3.1. An important requirement of our projection methods is to maintain these constraints. For example, in the bivariate case above, the matrix elements  $M(u)_{14} = M(u)_{22}$  are both equal to  $u_{20}$ . We now outline a simple algorithm to find a non-redundant matrix representation of these constraints in the general  $n$  variable case. To list these constraints we start from the first row and traverse the matrix from left to right across the rows and then traverse the rows from top to bottom. Note also that we only need to examine entries above the main diagonal since the matrix is symmetric.

For  $M(u)$  in Table 3.1 the first linear constraint traversing from the first row is  $M(u)_{14} = M(u)_{22}$ . We denote  $e_i$  as the  $i$ -th unit vector and  $E_{ij} = \frac{1}{2}(e_i^T e_j + e_j^T e_i)$  as the  $ij$ -th unit matrix. To impose this first constraint on a matrix  $M \in \mathcal{S}_+^{k+1}$ , we construct matrix  $A_2 = E_{22} - E_{14}$ . The constraint is then given by

$$\langle A_2, M \rangle = \text{trace}((E_{22} - E_{14})M) = 0.$$

Since we always assume  $M(u)_{1,1} = 1$ , we need to set  $A_1 = E_{11}$ . We can similarly construct  $A_3, A_4, \dots, A_r$ , where  $r$  is the number of the total linear constraints. We denote  $A_t$  the *matrix representative* of the  $t$ -th linear constraint.

---

**Algorithm 3.5.1:** Matrix representation of moment matrix constraints

---

```

1 Input( $d, n$ ) ( $d$  is the degree,  $n$  is the number of the variables) ;
2 Compute  $k := N(n, d) - 1 = \binom{d+n}{d} - 1$ .
3 Initialize an array  $T = \langle \alpha_{(\leq d)} \rangle$  of length  $k + 1$ ,  $T(i)$  is the  $i$ -th element of  $T$ .
4 Initialize an array  $S = \langle s \rangle$  of length  $k + 1$  with the  $i$ -th element  $S(i) = [(1, i); T(i)]$ .
5 Let  $t = 2$  and  $A_1 = E_{11}$ . for  $i$  from 2 to  $k + 1$ , do
6   for  $j$  from  $i$  to  $k + 1$ , do
7     if  $\exists g, h, \alpha$  with  $s = [(g, h); \alpha] \in S$  such that  $T(i) + T(j) = \alpha$  then
8        $A_t = E_{ij} - E_{gh}$ ,  $t = t + 1$ 
9     else
10      Adjoin a new element  $s = [(i, j); \alpha]$  to  $S$  where  $\alpha = T(i) + T(j)$ 
11    end if
12  end for
13 end for
14 Output( Return an array of  $(k + 1) \times (k + 1)$  matrix representatives  $\{A_t\}$  where  $t \in \mathcal{E}$ ,
       $\mathcal{E} = \{1, 2, \dots, r\}$  and  $r$  is the total number of the linear constraints.);

```

---

Algorithm 3.5.1 determines all the (non-redundant) matrix representatives of the linear constraints of the multivariate moment matrix. For example, if the input is  $(d, n) = (2, 2)$ , then  $T = [(0, 0), (1, 0), (0, 1), (2, 0), (1, 1), (0, 2)]$  and

$$S = [[(1, 1); (0, 0)], [(1, 2); (1, 0)], \dots, [(1, 6); (0, 2)]]$$

There are no redundant constraints produced by this algorithm. This avoids having an overdetermined linear system.

### 3.5.2 First step of facial reduction

Semidefinite programming has become an important tool in many areas of optimization and algebraic geometry, e.g., [52, 9, 2]. The semidefinite cone  $\mathcal{S}_+^t$  has been extensively studied and the facial structure is well understood. If  $X \in \mathcal{S}_+^t$ , then we let  $\text{face}(X, \mathcal{S}_+^t)$  denote the smallest face of  $\mathcal{S}_+^t$  containing  $X$ . And if  $f$  is a face of  $\mathcal{S}_+^t$ , denoted  $f \trianglelefteq \mathcal{S}_+^t$ , then the *conjugate face* is  $f^c := f^\perp \cap \mathcal{S}_+^t$ . Let  $X = \begin{bmatrix} U & V \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U & V \end{bmatrix}^T$  be the spectral decomposition of  $X$  with  $\begin{bmatrix} U & V \end{bmatrix}$  orthogonal and both  $D \in \mathcal{S}_{++}^r$  and diagonal. Then

$$\begin{aligned} \text{face}(X, \mathcal{S}_+^t) &= US_+^r U^T \\ &= \{Y \in \mathcal{S}_+^t : V^T Y = 0\} \\ &= \{Y \in \mathcal{S}_+^t : \text{trace}(VV^T)Y = 0\}. \end{aligned}$$

Similarly,

$$\begin{aligned} \text{face}(X, \mathcal{S}_+^t)^c &= VS_+^{t-r} V^T \\ &= \{Z \in \mathcal{S}_+^t : U^T Z = 0\} \\ &= \{Z \in \mathcal{S}_+^t : \text{trace}(UU^T)Z = 0\}. \end{aligned}$$

**Problem 3.5.2 (Moment Matrix Feasibility Problem)** *Our main problem is the following feasibility problem for the moment matrix  $M$ :*

$$\mathcal{A}(M) = b = e_1, \quad B^T M = 0, \quad M \in \mathcal{S}_+^{k+1}, \quad (3.11)$$

Here  $k$  and the linear transformation  $\mathcal{A}$  is obtained from Algorithm 3.5.1.  $\mathcal{A}(M) = \left( \langle A_t, M \rangle \right)_{t \in \mathcal{E}} \in \mathbb{R}^{r \times 1}$ . The full column rank matrix  $B$  is obtained from the coefficient matrix in Definition 3.2.1 and equation (3.6).

The following Theorem 3.5.1 provides the details of the system after 1 step of facial reduction obtained by applying the coefficient matrix constraint to the moment matrix, i.e.,  $B^T M = 0$ . Recall from Algorithm 3.5.1, we get an array of representing matrix  $A_t$ 's where  $t \in \mathcal{E}$ ,  $\mathcal{E} = \{1, 2, \dots, r\}$ .



**Theorem 3.5.1 (First step facial reduction)** *Let  $B \in \mathbb{R}^{N(n,d) \times m}$  be as above and of full column rank. Let  $V \in \mathbb{R}^{N(n,d) \times (N(n,d)-m)}$  satisfy  $V^T B = 0$  and  $\begin{bmatrix} B & V \end{bmatrix}$  nonsingular. Let*

$$\bar{A}_t := V^T A_t V, \quad \forall t \in \mathcal{E} = \{1, 2, \dots, r\}$$

*and define the linear transformation  $\bar{\mathcal{A}} : \mathcal{S}^{N(n,d)-m} \rightarrow \mathbb{R}^{r \times 1}$  by*

$$\bar{\mathcal{A}}(P) := \left( \langle \bar{A}_t, P \rangle \right)_{t \in \mathcal{E}}. \quad (3.12)$$

*Then Problem 3.5.2 is equivalent to*

$$\bar{\mathcal{A}}(P) = b, \quad P \in \mathcal{S}_+^{N(n,d)-m}, \quad (3.13)$$

*where we can recover the moment matrix using  $M = V P V^T$ .*

**Proof** It can be proved easily using the property of the trace product. ■

Note that for stability, we need to process the linear constraint (3.12) further to obtain an equivalent linear system  $\hat{\mathcal{A}}(\hat{P}) = \hat{b}$  where  $\hat{A}$  is an onto map.

### Potential second facial reduction

Our initial semidefinite moment problem is a feasibility problem of the form

$$B^T M(u) = 0, \quad M(u) \geq 0, \quad (3.14)$$

where  $B$  is a given coefficient matrix and the moment matrix  $M(u)$  is a linear function of the variables  $u$ . Constraints on  $M(u)$  are described in Section 3.5.1. In Section 3.5.3 the problem is changed to equality form and then uses facial reduction to get the form

$$\bar{\mathcal{A}}(P) = b, \quad P \geq 0. \quad (3.15)$$

This form includes the first step of facial reduction using the matrix  $B$ , see Theorem 3.5.1 and (3.12).

The projection methods behave poorly, converge slowly, when the Slater condition fails, e.g., [20]. We therefore attempt to apply further steps of facial reduction and reduce system

(3.15) until a strictly feasible point exists. We use the following theorem of the alternative or characterization of a strictly feasible point. (See e.g., [15].)

$$\begin{aligned} \exists P, \bar{\mathcal{A}}(P) = b, P > 0 \\ \iff \\ Z = \bar{\mathcal{A}}^* y \geq 0, b^T y = 0 \implies Z = 0. \end{aligned} \tag{3.16}$$

Note that if a  $Z \neq 0$  can be found satisfying the left part of the bottom half of (3.16) and for the top half  $P \geq 0, \bar{\mathcal{A}}(P) = b$ , then

$$0 = b^T y = \langle \bar{\mathcal{A}}(P), y \rangle = \langle P, Z \rangle \implies PZ = 0 \implies \text{range } P \subseteq \text{null } Z.$$

Therefore, if the full column rank matrix  $W$  satisfies  $\text{range } W = \text{null } Z$ , then we can facially reduce the problem to a lower matrix  $\bar{P}$  using the substitution  $P = W\bar{P}W^T$ , i.e., we can restrict the feasibility problem in (3.15) to the face  $W\bar{\mathcal{S}}_+W^T$ .

We can implement the test in (3.16) in several ways. One way is to solve the following minimization problem <sup>2</sup>

$$\begin{aligned} p^* := \min \quad & \frac{1}{2}(\bar{b}^T y)^2 \\ \text{s.t.} \quad & Z = \bar{\mathcal{A}}^* y \geq 0 \\ & \text{trace } \bar{\mathcal{A}}^* y = 1 \end{aligned}$$

where

$$\bar{\mathcal{A}}^* y = \sum_{t=1}^r (\bar{A}_t y).$$

If the objective  $p^*$  is 0, then it implies we may need a second facial reduction. A *stable* approach, in the sense that strict feasibility holds, to solving this auxiliary problem is given in [15] as

$$\begin{aligned} \max \quad & \delta \\ \text{s.t.} \quad & Z = \bar{\mathcal{A}}^* y \geq \delta I \\ & \text{trace } Z = 1 \\ & \bar{b}^T y = 0 \end{aligned} \tag{3.17}$$

---

<sup>2</sup> This can be implemented in e.g., CVX using the *norm* function or absolute value function for the objective, i.e., we minimize  $|\bar{b}^T y|$  rather than using the squared term.

### Backward stability for facial reduction steps

We now see that we can find the equivalent facial reduced problem efficiently and accurately. We start with the Moment Matrix Feasibility Problem in (3.11).

$$\mathcal{A}(M) = b = e_1, \quad B^T M = 0, \quad M \in \mathcal{S}_+^{N(n,d)}.$$

As above,  $B \in \mathbb{R}^{(k+1) \times m}$  and is full column rank. We apply the QR factorization and numerically obtain the output  $B \approx \tilde{Q}\tilde{R}$ , where  $Q = \begin{bmatrix} \tilde{U} & \tilde{V} \end{bmatrix}$  is orthogonal, and  $\tilde{R}$  upper triangular with the last  $m$  rows being zero, see e.g., [27]. The QR factorization is backwards stable, i.e., we get the exact equation

$$\tilde{Q}\tilde{R} = B + \delta B, \quad \frac{\|\delta B\|}{\|B\|} = O(\epsilon_{\text{machine}}), \quad (3.18)$$

Thus we have *exactly* found the QR factorization of a nearby matrix. We then use Theorem 3.5.1 to obtain the facially reduced problem in (3.13) i.e., we form the matrices  $\tilde{A}_t$ . The matrix  $V$  has orthonormal columns. Therefore the congruence is a backward stable operation and we have

$$\tilde{A}_t = \tilde{V}^T (A_t + \delta A_t) \tilde{V}, \quad \frac{\|\delta A_t\|}{\|A_t\|} = O(\epsilon_{\text{machine}}), \quad \forall t \in \mathcal{E} = \{1, 2, \dots, r\}. \quad (3.19)$$

Therefore, we can combine the above two steps and conclude that the first step of facial reduction is a stable operation, i.e.,

$$\tilde{\mathcal{A}}(P) = b, \quad P \in \mathcal{S}_+^{N(n,d)-m}, \quad (3.20)$$

is obtained efficiently and accurately; we have found the *exact* facial reduction of a nearby problem.

Note that we then use a singular value decomposition to remove the redundant linear constraints so that the linear map  $\tilde{A}$  in the resulting linear constraints can be assumed to be onto. This can be done using the SVD factorization, again a backwards stable algorithm. We have shown the following.

**Theorem 3.5.2 (Backward stability of first FR)** *The first step of facial reduction is backward stable. More precisely, we find a linear system (3.20) with  $\tilde{A}$  onto and equivalent to a nearby system to the original moment matrix feasibility problem in the sense of (3.18) and (3.19). ■*

We do not include the analysis for a second step of facial reduction. This is more difficult as we need to include the accuracy in solving the auxiliary problem for the theorem of the alternative discussed in Section 3.5.2. Such an analysis can be found in [15, Theorem 1.38].

### 3.5.3 Projection methods

We now consider two projection methods. We first consider the method of *alternating projection*, *MAP* and use the defined projections to introduce the *Douglas-Rachford reflection-projection* method. It is the latter method that we implement as it displayed better convergence properties in our tests.

#### Method of alternating projections, MAP

The method of alternating projections, MAP, is particularly simple, see e.g., the recent book [24]. Let  $\text{s2vec}$  denote the mapping (isometry) from a matrix to a column vector taken columnwise with the off-diagonal elements multiplied by  $\sqrt{2}$ . Let  $\text{s2Mat} = \text{s2vec}^* = \text{s2vec}^{-1}$  be the inverse mapping from a column vector to a matrix. The inverse here is identical to the *adjoint map*. Let  $L = (\text{s2vec}(\bar{A}_t)^T)_{t \in \mathcal{E}}$  denote the matrix representation for  $\bar{\mathcal{A}}$  in Theorem 3.5.1 ( $\text{s2vec}(\bar{A}_t)^T$  is the  $t$ -th row of  $L$ ).

We begin with an initial estimate, e.g.,  $P_c = \alpha I \in \mathcal{S}_+^{N(n,d)-m}$  for a *large*  $\alpha > 0$ . There are two projections we use to update the current point  $P_c$ . First, we look at  $\mathcal{P}_{\mathcal{L}}$ , the *linear manifold projection*. We map  $P_c$  to a column vector  $p_c = \text{s2vec}(P_c)$ , then for the linear system  $Lp = b = e_1$  where  $L$  has full row rank, we solve the nearest point problem  $\min \left\{ \frac{1}{2} \|p - p_c\|_2^2 : Lp = b \right\}$ , i.e., we find the projection onto the linear manifold for the linear constraints. We use  $L^\dagger$ , the *Moore-Penrose generalized inverse* of  $L$ . The residual and the  $p_l$  satisfying the minimization problem are then

$$r_c = b - Lp_c; \quad p_l = p_c + L^\dagger r_c. \quad (3.21)$$

Second, we project the updated symmetric matrix  $P_L = \mathcal{P}_{\mathcal{L}}(P_c) = \text{s2Mat}(p_l)$  onto the semidefinite cone using the Eckart-Young Theorem [22], i.e., we diagonalize and zero out the negative eigenvalues. We denote  $\mathcal{P}_{S_+}$ , the *positive semidefinite projection* and get the new positive semidefinite approximation  $\mathcal{P}_{S_+}(P_L)$ .

We repeat the projection steps in Items 1, 2, 3 described below till a sufficiently small desired tolerance is obtained in the norm of the residual.

1. Evaluate the residual  $r_c = b - Lp_c$ . Use the residual to evaluate the linear projection and obtain the update

$$P_L = \mathcal{P}_{\mathcal{L}}(P_c).$$

2. Evaluate the positive semidefinite projection using the Eckart-Young Theorem and update the current approximation

$$P_{PSD} = \mathcal{P}_{S_+}(P_L).$$

3. Update the cosine value in (3.22). Then update  $P_c = P_{PSD}$ .

The (linear) convergence rate is measured using cosines of angles from three consecutive iterates

$$\cos(\theta) = \left( \frac{\text{trace}((P_L - P_c)^*(P_{PSD} - P_L))}{\|P_L - P_c\| \|P_{PSD} - P_L\|} \right). \quad (3.22)$$

### Douglas-Rachford reflection method

Recall the projections defined above  $\mathcal{P}_{\mathcal{L}}, \mathcal{P}_{S_+}, P_{PSD}$ . We want to find, see (3.13),

$$P \in \mathcal{G} \cap \mathcal{S}_+^{N(n,d)-m}, \quad \text{where } \mathcal{G} := \{P : \tilde{\mathcal{A}}(P) = b = R\}.$$

We now apply the Douglas-Rachford (DR) projection/reflection method [18]. (See also e.g., [3, 12, 34, 6].)

Using the QR algorithm applied to  $B$  to find  $V$  and  $\tilde{\mathcal{A}}$ , we start with an initial estimate

$$P_0 = \alpha I \in \mathcal{S}_+^{N(n,d)-m} \text{ for some } \alpha. \quad (3.23)$$

Define the *reflections*  $\mathcal{R}_{\mathcal{L}}, \mathcal{R}_{PSD} : \mathcal{S}_+^{N(n,d)-m} \rightarrow \mathcal{S}_+^{N(n,d)-m}$  using the corresponding projections, i.e.,

$$\mathcal{R}_{\mathcal{L}}(P) := 2\mathcal{P}_{\mathcal{L}}(P) - P, \quad \mathcal{R}_{PSD}(P) := 2\mathcal{P}_{S_+}(P) - P.$$

- **Initialization:** We set our current estimate  $P_c = P_0$ . We calculate the residual  $Res_{\mathcal{L}} = R - \tilde{\mathcal{A}}(P_c)$ , set  $normres = \|Res_{\mathcal{L}}\|$ , denote the reflected residual  $Resrefl_{\mathcal{L}} = Res_{\mathcal{L}}$  and reflected point  $\mathcal{R}_{PSD} = P_c$ .

- **Iterate:** We continue iterating from this point while  $normres > toler$ , our desired tolerance.
- 1. We use  $Resrefl_{\mathcal{L}}$  to project the current reflected PSD point  $\mathcal{R}_{PSD}$  onto the linear manifold to get the projected point  $P_{\mathcal{L}} = \mathcal{R}_{PSD} + s2Mat(L^{\dagger}Resrefl_{\mathcal{L}})$ . Then we reflect to get our second reflection point  $\mathcal{R}_{\mathcal{L}} = 2P_{\mathcal{L}} - \mathcal{R}_{PSD}$ .
- 2. At this time we set our new/current estimate for convergence to be  $P_c = P_{new} = (P_c + \mathcal{R}_{\mathcal{L}})/2$ .
- 3. We now project  $P_c$  to get  $P_{PSD} = \mathcal{P}_{S_+}(P_c)$ . We check the residual here for the stopping criteria  $normres = \|Res_{\mathcal{L}}\| = \|R - \tilde{\mathcal{A}}(P_{PSD})\|$ .
- 4. We now calculate the first reflection point  $\mathcal{R}_{PSD} = 2P_{PSD} - P_c$  and update the reflected residual  $Resrefl_{\mathcal{L}} = R - \tilde{\mathcal{A}}(\mathcal{R}_{PSD})$ .

Also according to the basic theorem on the convergence of the sequence, [12, Thm 3.3, Page 11], the residuals of the projections of the iterates on one of the sets have to be used for the stopping criteria. We use the residual after the projection onto the SDP cone since we want our final matrix to be semidefinite.

Algorithm 3.5.2 summarizes our Facial reduction & Douglas-Rachford method.

---

**Algorithm 3.5.2:** FDR method

---

- 1 Input( Degree of system  $d$ , number of variables  $n$ , a  $N(n, d) \times m$  coefficient matrix  $B$ ) ;
  - 2 Compute the matrix representation  $A$  using Algorithm 3.5.1.;
  - 3 Use QR to find  $V$  s.t.  $V^T B = 0$  and  $\begin{bmatrix} B & V \end{bmatrix}$  nonsingular; compute the matrix representation  $L$  of the linear transformation  $\tilde{\mathcal{A}}$  described in Theorem 3.5.1.;
  - 4 Start at an initial point  $P_0$  satisfying (3.23).;
  - 5 Iterate:  $P_{j+1} = \frac{1}{2}(P_j + \mathcal{R}_{PSD}(\mathcal{R}_{\mathcal{L}}(P_j)))$ , for all  $j = 0, 1, \dots$ ;
  - 6 Stop if  $normres \leq toler$ .;
  - 7 Output( A PSD  $N(n, d) \times N(n, d)$  moment matrix  $M = VP_{j+1}V^T$ .)
- 

Our empirical studies showed that the Douglas-Rachford approach outperformed MAP and

also outperformed the SeDuMi interior point method within the YALMIP toolbox. Though the Douglas-Rachford iteration has only a linear convergence rate, the method converged robustly to the intersection of the linear constraints and the semidefinite cone. We note that for two subspaces, the linear rate for the method is given by the cosine of the Friedrichs angle between them, see e.g., [5, 7]. Details on the numerical tests follow.

## 3.6 Numerical experiments

In this section we present the numerical tests for the GIF-Moment Matrix Algorithm 3.4.1 that combines the Geometric Involutive Form with an SDP solver. We consider the two SDP feasibility solving algorithms: the FDR Algorithm 3.5.2 with facial reduction and the standard interior point solver SeDuMi but without facial reduction. GIF is combined with the two SDP approaches to yield GIF-FDR and GIF-SeDuMi, respectively.

In Section 3.6.1 we consider a class of random univariate polynomials with varying degree  $d$ . The results are displayed in Figure 3.1 on page 76, and Figure 3.2 on page 77. Results for the examples given in Sections 3.6.2 and 3.6.3 are summarized in Table 3.2 page 82.

We used MATLAB version 2014a and Maple version 18. The computations were carried out on a desktop with ubuntu 12.04 LTS, Intel Core™2 Quad CPU Q9550 @ 2.83 GHz  $\times$  4, 8GB RAM, 64-bit OS, x64-based processor.

### 3.6.1 A class of random univariate polynomials

We first consider root finding for polynomials of the form

$$p_d(x) = a_{d,0} + a_{d,1} x + a_{d,2} x^2 + \cdots + a_{d,d} x^d, \quad d = 1, 3, 5, \dots \quad (3.24)$$

where  $a_{d,j} \sim N(0, 1)$ . A famous early work on random polynomials such as (3.24) is given by Kac in [29] who derived an integral formula for the average number of real roots of  $p_d(x)$ :

$$E_d = \frac{4}{\pi} \int_0^1 \sqrt{\frac{1}{(1-t^2)^2} - \frac{(d+1)^2 t^{2d}}{(1-t^{2d+2})^2}} dt. \quad (3.25)$$

An asymptotic form for large  $d$  was determined to be  $E_d \approx \frac{2}{\pi} \log(d) + 0.6257358072\dots + \frac{2}{\pi d} + O\left(\frac{1}{d^2}\right)$ , e.g., [23] and the references therein.

We applied GIF-FDR and GIF-SeDuMi to the random polynomials  $p_d(x)$  for odd degrees  $d$  with  $3 \leq d \leq 51$ . For each odd degree  $j$ , 10 sample random polynomials were generated by selecting their coefficients as independent samples from  $N(0, 1)$ . Algorithms GIF-FDR and GIF-SeDuMi were then applied to approximate the minimal polynomial generating their real radical. The residual error for each polynomial at odd degree  $j$  was computed by substituting that roots of the minimal polynomial into the original input polynomial  $|p_j|$ . The average of the  $\log_{10}$  of all these 10 residual errors was computed for each degree  $j$ . We also checked that the mean number of the real roots of these samples was approximately given by (3.25).

We report on the comparison of the average residual errors versus degree in Figure 3.1. It is clear that GIF-FDR consistently obtains significantly better accuracy than GIF-SeDuMi. Figure 3.1 also contains comparison for cpu-time. Each instance was solved by GIF-SeDuMi first and the residual error recorded. This error was then used for the desired residual error when applying GIF-FDR. The average cpu-times per degree are plotted. Again we see that GIF-FDR performed consistently better even though it has a theoretical linear convergence time whereas interior point methods have a theoretical superlinear convergence time. In Figure

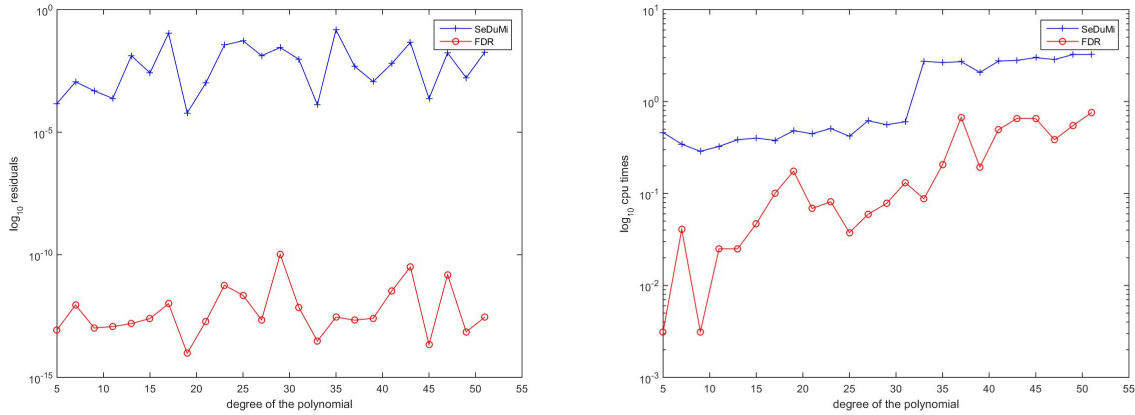


Figure 3.1: Comparison in residual and cputime of GIF-FDR vs GIF-SeDuMi for random polynomials  $p_d(x) = \sum_1^d a_{d,j} x^j$  at odd degrees  $3 \leq d \leq 51$  with  $a_{d,j} \sim N(0, 1)$ .

3.2 we used the popular performance profile approach [17] with the following performance profile function

$$\rho_s(\tau) = \frac{\text{size}\{p \in \mathcal{P} : r_{p,s} \leq \tau\}}{\text{size}(\mathcal{P})}, \quad s = 1, 2 \quad (3.26)$$



where  $\mathcal{P}$  is the set of problems and  $r_{p,s}$  is the ratio of the performance of solver  $s$  to the best performance by any solver on this problem  $p$ . These figures show FDR ( $s = 2$ ) has outperformed SeDuMi ( $s = 1$ ) in residual and cputime.

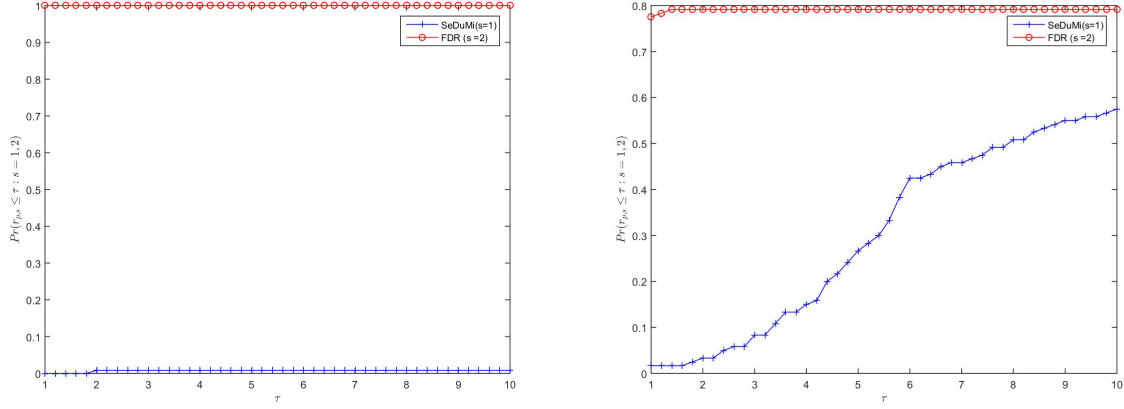


Figure 3.2: Performance profile of GIF-FDR vs GIF-SeDuMi for random polynomials  $p_d(x) = \sum_1^d a_{d,j} x^j$  at each odd degrees  $3 \leq d \leq 51$  with  $a_{d,j} \sim N(0, 1)$ . The profile function used is (3.26).

### 3.6.2 Examples of Ma, Wang and Zhi [37]

Ma, Wang and Zhi [37, 36] present an approach using Pommaret Bases coupled with moment matrix completion to approximate the real radical ideal of a polynomial variety. We applied our approach to [37, Examples 4.1-4.6], with the results shown in Table 3.2. In each of the examples we first applied GIF-FDR and then GIF-SeDuMi (i.e., FDR replaced with SeDuMi SDP solver). In each case we obtained a geometric involutive basis which can be independently verified as a geometric involutive basis for the real radical. In [37] Pommaret bases are successfully obtained for the real radical for these examples. For an additional verification, we took the polynomials resulting from the final moment matrix from GIF-FDR, and summed their squares. Then we found an approximation to the roots by finding the minimum of this polynomial using the MATLAB optimization toolbox. Finally, we substituted these approximate roots into the original input polynomials and evaluated the residual error. The results in the final column in Table 3.2 show a small residual error.

Here are the 6 systems of polynomials corresponding to the examples in [37]:

$$\{x_1^2 + x_1x_2 - x_1x_3 - x_1 - x_2 + x_3, \quad x_1x_2 + x_2^2 - x_2x_3 - x_1 - x_2 + x_3, \\ x_1x_3 + x_2x_3 - x_3^2 - x_1 - x_2 + x_3\} \quad (3.27a)$$

$$\{x_1^2 - x_2, \quad x_1x_2 - x_3\} \quad (3.27b)$$

$$\{x_1^2 + x_2^2 + x_3^2 - 2, \quad x_1^2 + x_2^2 - x_3\} \quad (3.27c)$$

$$\{x_3^2 + x_2x_3 - x_1^2, \quad x_1x_3 + x_1x_2 - x_3, \quad x_2x_3 + x_2^2 + x_1^2 - x_1\} \quad (3.27d)$$

$$\{(x_1 - x_2)(x_1 + x_2)^2(x_1 + x_2^2 + x_2), \quad (x_1 - x_2)(x_1 + x_2)^2(x_1^2 + x_2^2)\} \quad (3.27e)$$

$$\{(x_1 - x_2)(x_1 + x_2)(x_1 + x_2^2 + x_2), \quad (x_1 - x_2)(x_1 + x_2)(x_1^2 + x_2^2), \\ x_1 \geq 1, \quad x_2 \geq 1\} \quad (3.27f)$$

**System (3.27a) for [37, Example 4.1]:** The first step of applying Algorithm 3.4.1 is to use Maple and apply the GIF Algorithm 3.3.1, page 62, with input tolerance  $10^{-10}$  to (3.27a). This shows that the system is already in geometric involutive form. The corresponding Pommaret basis is given in [37, Example 4.1]. The Pommaret basis looks different from the system, but is just a linear combination of the system's polynomials to accomplish the Gröbner-like requirement for its highest terms under the term ordering prescribed in the problem. The resulting coefficient matrix of this GIF form, is a full rank  $m = 3$ ,  $3 \times 10$  matrix which is input to the FDR algorithm. The dimension of the kernel for GIF form is  $d = 7$ . Since the coefficient matrix has rank  $m = 3$ , one facial reduction yields a reduced  $(10 - m) \times (10 - m) = 7 \times 7$  moment matrix. Application of the FDR algorithm yields convergence in 2 iterations and 0.02 secs, with a projected residual error of  $10^{-15}$ . These statistics are shown in Table 3.2. The output of FDR is a full  $10 \times 10$  moment matrix of rank  $r = 7$ . Since  $d = 7 = r$ , Algorithm 3.4.1 terminates with the input system as its output. It can be checked that the ideal generated by this system is real radical.

For comparison, application of GIF-SeDuMi to (3.27a) using a tolerance of  $10^{-10}$  in Maple resulted in a residual error of  $10^{-10}$ , as listed in the last column of Table 3.2, and an approximation of the generators of the real radical.

**System (3.27d) for [37, Example 4.4]:** This is very similar to the previous system (3.27a). As [37] notes the coordinates for this example are not delta-regular, which they and we remedy by

a linear change of coordinates. We show that the original system is geometrically involutive, which is equivalent to the determination of a Pommaret basis by [37]. Just as in the previous example, we form a  $10 \times 10$  moment matrix from the GIF form, which is transformed by one facial reduction to a  $7 \times 7$  matrix. There are no additional facial reductions, and the full moment matrix and its rank  $r$  are determined. We find that dimension of the kernel for GIF form is  $d = 7 = r$ , so Algorithm 3.4.1 terminates with the input system as its output. It can be verified the the output is a GIF form for the real radical of the ideal.

Application of GIF-SeDuMi to (3.27d) using a tolerance of  $10^{-8}$  in Maple resulted in a residual error of  $10^{-8}$  and an approximation of the generators of the real radical.

**System (3.27b) for [37, Example 4.2]:** This is quite similar to the systems (3.27b) and (3.27d). Our methods are similarly efficiently applied to this system. Our GIF algorithm first applied one prolongation to the second system (3.27b) to yield a degree 3 system. After projecting from this degree 3 system it shows that the resulting degree 2 system is involutive and consists of 3 polynomials. This degree 2 system is geometrically equivalent to the Pommaret basis found by [37]. This system is simply the original 2 polynomials, together with their compatibility condition or S-polynomial  $x_2(x_1^2 - x_2) - x_1(x_1x_2 - x_3) = x_1x_3 - x_2^2$ . Thus the input system  $R$  is replaced with  $\pi DR$  represented by its  $3 \times 10$  coefficient matrix. The resulting  $10 \times 10$  moment matrix is facially reduced to a  $7 \times 7$  moment matrix. As in the previous examples, no new relations are detected in the kernel of the output matrix of the FDR method,  $d = r = 7$  and the algorithm terminates. It can be verified that the GIF form is a basis for the real radical ideal of the input system.

Application of GIF-SeDuMi to (3.27b) using a tolerance of  $10^{-9}$  in Maple resulted in a residual error of  $10^{-9}$  and an approximation of the generators of the real radical.

Unlike the systems (3.27a),(3.27b),(3.27d), the remaining three systems (3.27c),(3.27e),(3.27f) of [37] lead to new members in the kernel of their moment matrices.

**System (3.27c) for [37, Example 4.3]:** Our initial application of FDR showed slow convergence. However a random linear change of coordinates applied to the input system  $R$  dramatically improved the convergence. Applying the GIF algorithm we found that  $\hat{D}R$  is involutive and has a  $8 \times 20$  coefficient matrix. The dimension of its kernel is  $d = 12$ . Applying the FDR algorithm, we obtain a PSD moment matrix with rank  $r = 7 \neq d$  so the algorithm

has not terminated. The new member of the real radical arising in the moment matrix kernel can be alternatively derived by hand by elimination of two of the systems polynomials:  $x_1^2 + x_2^2 + x_3^2 - 2 - (x_1^2 + x_2^2 - x_3) = x_3^2 + x_3 - 2 = (x_3 + 2)(x_3 - 1)$ . Then noting, as explained in [37], that only the root  $x_3 = 1$  leads to real solutions. The GIF form of the new system from the kernel of the moment matrix is computed which has degree 2. Its coefficient matrix is  $5 \times 10$  and has kernel of dimension  $d = 5$ . After applying FDR algorithm, the second PSD moment matrix then was computed quickly and accurately as a  $10 \times 10$  matrix. The rank of the second moment matrix is  $r = 5 = d$ , so our algorithm has terminated. It can be checked that the output is equivalent to that found by [37] and that the resulting GIF form is a basis for the real radical.

Application of GIF-SeDuMi to (3.27c) using a tolerance of  $10^{-8}$  in Maple resulted in a residual error of  $10^{-9}$  and an approximation of the generators of the real radical.

**System (3.27e) for [37, Example 4.5]:** Direct application of Algorithm 3.4.1 to (3.27e) is relatively inefficient. Instead of this approach we consider an alternative subsystem approach which has the potential to be applied to larger systems. Exploiting subsystem structure is a long established approach in system solving.

We apply Algorithm 3.4.1 to the subsystem consisting of the first polynomial of  $P_1 = (x_1 - x_2)(x_1 + x_2)^2(x_1 + x_2^2 + x_2)$  of (3.27e). The GIF form of  $P_1$  is just  $P_1$ , and its coefficient matrix is  $1 \times 21$  matrix with a kernel of dimension  $d = 20$ . The corresponding moment matrix is  $21 \times 21$ , which is reduced to a  $20 \times 20$  matrix after one facial reduction. It has rank  $r = 18 \neq d$ . So the algorithm has not terminated, and new members of the real radical are identified from the kernel of the moment matrix. The new system is degree 5 and has 3 polynomials. Algorithm GIF shows that the first projection of this system is involutive and is a single fourth degree polynomial. Its coefficient matrix is  $1 \times 15$  and its kernel has dimension  $d = 14$ . The FDR algorithm produces a  $15 \times 15$  positive semidefinite moment matrix with the rank being  $r = 14 = d$ . The algorithm terminates to coefficient errors within  $10^{-10}$  with output as a single polynomial which is approximately:

$$(x_1 - x_2)(x_1 + x_2)(x_1 + x_2^2 + x_2) \quad (3.28)$$

It can be checked that (3.28) is a geometric involutive basis for the real radical for the ideal generated by  $P_1$ .

Similarly we apply Algorithm 3.4.1 to the second polynomial of (3.27e) which is given by  $P_2 = (x_1 - x_2)(x_1 + x_2)^2(x_1^2 + x_2^2)$ . The algorithm now terminates with output as a single polynomial which is approximately:

$$(x_1 - x_2)(x_1 + x_2) \quad (3.29)$$

This can be verified to be a geometric involutive basis for the real radical of the ideal generated by  $P_2$ .

Then we consider the system

$$(x_1 - x_2)(x_1 + x_2)(x_1 + x_2^2 + x_2), (x_1 - x_2)(x_1 + x_2) \quad (3.30)$$

Application of GIF to (3.30) reduces it to a geometric involutive basis which is approximately

$$(x_1^2 - x_2^2) \quad (3.31)$$

A further application of FDR reveals that (3.31) is a GIF form for the real radical of the ideal of (3.27e).

Application of GIF-SeDuMi to (3.27e) also yields an approximation of the generators of the real radical. The most notable feature of this calculation was the its requirement of fairly large tolerances ( $10^{-4}$  and  $10^{-5}$ ). Reference [37, Example 4.5] also notes a similarly large tolerance in their calculations, to correctly compute the real radical for this example.

**System (3.27f) for [37, Example 4.6]:** Let  $Q_1 = \{(x_1 - x_2)(x_1 + x_2)(x_1 + x_2^2 + x_2), (x_1 - x_2)(x_1 + x_2)(x_1^2 + x_2^2)\}$  then (3.27f) is  $Q_1$  subject to the constraints  $x_1 \geq 1, x_2 \geq 1$ .

Applying Algorithm 3.4.1 to  $Q_1$  yields a geometric involutive basis which is approximately  $x_1^2 - x_2^2$ . This can be independently verified to be a geometric basis for the real radical of  $Q_1$ . The statistics of this reduction are given in Table 3.2 in the row labeled as Ex 4.6  $Q_1$ .

To impose  $x_1 \geq 1, x_2 \geq 1$  we substitute  $x_1 = x_3^2 + 1, x_2 = x_4^2 + 1$  into the geometric involutive basis of the real radical of  $Q_1$ , that is into  $x_1^2 - x_2^2$ , and reduce the resulting polynomial  $Q_2 = (x_3^2 + 1)^2 - (x_4^2 + 1)^2 = (x_3^2 - x_4^2)(x_3^2 + x_4^2 + 2)$  with Algorithm 3.4.1 to yield a basis for its real radical which is  $x_3^2 - x_4^2$  or equivalently  $x_1 - x_2$  in agreement with [37, Example 4.6]. The statistics of this reduction are given in Table 3.2 in the row labeled as Ex 4.6  $Q_2$ .

	Input	FDR	FDR	FDR	Mom Mtx	GIF-SeDuMi
Polyn.	data	# its	cpu-sec	res-err	redn factor	Int Pt
System	(n,d,m)	(1,2)	(1,2)	max(1,2)	$s(M)/s(\hat{M})$	tol, res-err
Ex 4.1	(3,2,3)	2	0.02	$10^{-15}$	$\frac{10}{7}$	$10^{-10}, 10^{-10}$
Ex 4.2	(3,2,2)	156	0.23	$10^{-14}$	$\frac{10}{7}$	$10^{-9}, 10^{-9}$
Ex 4.3	(3,2,2)	256, 2	2.4, 0.08	$10^{-13}$	$\frac{20}{12}, \frac{10}{5}$	$10^{-8}, 10^{-9}$ —
Ex 4.4	(3,2,3)	106	0.06	$10^{-15}$	$\frac{10}{7}$	$10^{-8}, 10^{-8}$
Ex 4.5 $P_1$	(2,5,1)	9582, 29	7.0, 0.17	$10^{-13}$	$\frac{21}{20}, \frac{15}{14}$	$10^{-4}, 10^{-8}$
Ex 4.5 $P_2$	(2,5,1)	148, 1	0.3, 0.06	$10^{-14}$	$\frac{21}{20}, \frac{6}{5}$	$10^{-5}, 10^{-8}$
Ex 4.6 $Q_1$	(2,4,2)	34, 2	0.11, 0.08	$10^{-13}$	$\frac{21}{15}, \frac{6}{5}$	$10^{-6}, 10^{-8}$
Ex 4.6 $Q_2$	(2,4,1)	86, 1	0.28, 0.03	$10^{-14}$	$\frac{15}{14}, \frac{6}{5}$	$10^{-8}, 10^{-9}$
Cyl2d	(2,2,1)	1	0.06	$10^{-15}$	$\frac{6}{5}$	$10^{-10}, 10^{-13}$
Cyl3d	(3,2,2)	2	0.09	$10^{-15}$	$\frac{20}{12}$	$10^{-8}, 10^{-9}$
Cyl4d	(4,2,3)	7	0.31	$10^{-14}$	$\frac{70}{28}$	$10^{-7}, 10^{-8}$
Cyl5d	(5,2,4)	10	0.52	$10^{-14}$	$\frac{252}{64}$	DNC

Table 3.2: **Statistics for the application of GIF-FDR and GIF-SeDuMi:** Ex 4.1-4.6 are 6 examples in MWZ [37]; Cyl2d-Cyl5d are cylinder examples;  $n$  number of variables;  $d$  maximum polynomial degree;  $m$  number of polynomials; in columns 3, 4, two entries (1,2) are included for the number of iterations and cpu-time if FDR is used twice in the example; And we take the max value in the residual error columns 5 and 8;  $(s(M), s(\hat{M}))$  is sizes of moment matrix  $M$  and facially reduced matrix  $\hat{M}$ , resp.; column 7 is the SVD tolerance for GIF and the residual error for the moment matrix using the Interior Point calculation with SeDuMi, DNC - Did Not Converge; the Maple SVD computations in GIF-FDR were executed with tolerance  $:= 10^{-10}$  and *Digits*  $:= 15$ , resp.

Application of GIF-SeDuMi to (3.27f) also yields an approximation of the real radical. The most notable feature of this calculation was the large tolerance  $10^{-6}$  and residual error for the reduction of  $Q_1$ .

### 3.6.3 Intersecting higher dimensional cylinders

Consider the systems of polynomials defining the intersection of  $n - 1$  cylinders in  $\mathbb{R}^n$

$$Cyl_{nd} := x_1^2 + x_2^2 - 1, x_1^2 + x_3^2 - 1, \dots, x_1^2 + x_n^2 - 1. \quad (3.32)$$

Application of the GIF algorithm to the systems  $Cyl_{nd}$  for  $n = 2, 3, 4, 5$  show that the systems become geometrically involutive after 0, 1, 2, 3 prolongations respectively. The GIF-FDR algorithm converges quickly and accurately (see Table 3.2). It can be independently determined that in each case it yields an geometric involutive basis for the real radical. However SeDuMi-GIF crashes after several hours on the largest system  $Cyl_{5d}$ .

Further it can be determined that the cylinders form a complete intersection and the length of the prolongation to make them involutive, can be determined from the symbol of the initial system [40]. The lower degree input systems (3.32) are geometrically formally integrable, and it would be interesting to develop methods based on such lower degree systems, to determine, whether one can rule out new members in the kernel of the moment matrix of the prolonged involutive system from such lower degree systems.

Recently certain critical point methods have been developed for determining witness points [54, 28] on real components of real polynomial systems. Indeed the method developed in [54] is successful in finding a point on every component, if the ideal is both real radical, and forms a regular sequence. Consequently for systems such as those above, the real radical is an important property for such solvers. The regular sequence requirement can be checked by dimension computation and can exploit a formally integrable system which has lower degree than the involutive system. Interesting related results are given in [38]. By experiment we found that the 0 dimensional systems for the critical points of (3.32) are also real radical and remarkably have no non-real roots. The number of real critical points corresponding to  $n = 2, 3, 4, 5$  can be determined to be 2, 4, 8, 16.

## 3.7 Conclusion

SDP feasibility problems typically involve the intersection of the convex cone of semidefinite matrices with a linear manifold. Their importance in applications has led to the development of many specific algorithms. However these feasibility problems are often marginally infeasible, i.e., they do not satisfy strict feasibility as is the case for our polynomial applications. Such problems are *ill-posed* and *ill-conditioned*.

The main contribution of this paper is to introduce facial reduction, for the class of SDP

problems arising from analysis and solution of systems of real polynomial equations for real solutions. Facial reduction yields an equivalent problem for which there are strictly feasible points and which, in addition, are smaller. Facial reduction also reduces the size of the moment matrices occurring in the application of SDP methods. For example the determination of a  $k \times k$  moment matrix for a problem with  $m$  linearly independent constraints is reduced to a  $(k - m) \times (k - m)$  moment matrix by one facial reduction. We use facial reduction with our MATLAB implementation of Douglas-Rachford iteration (our FDR method). In the case of only one constraint, say as in the case of univariate polynomials, one might expect that the improvement in convergence due to that facial reduction would be minor. However we present a class of random univariate polynomials, where one such facial reduction combined with DR iteration, yields the real radical much more efficiently than the standard interior point method in SeDuMi. The high accuracy required by facial reduction and also the ill-conditioning commonly encountered in numerical polynomial algebra [50] motivated us to implement Douglas-Rachford iteration.

A fundamental open problem is to generalize the work of [32, 49] to positive dimensional ideals. The algorithm of [37, 36] for a given input real polynomial system  $P$ , modulo the successful application of SDP methods at each of its steps, computes a Pommaret basis  $Q$ :

$$\sqrt[k]{\langle P \rangle_{\mathbb{R}}} \supseteq \langle Q \rangle_{\mathbb{R}} \supseteq \langle P \rangle_{\mathbb{R}} \quad (3.33)$$

and would provide a solution to this open problem if it is proved that  $\langle Q \rangle_{\mathbb{R}} = \sqrt[k]{\langle P \rangle_{\mathbb{R}}}$ . We believe that the work [37, 36] establishes an important feature – involutivity – that will necessarily be a main condition of any theorem and algorithm characterizing the real radical. Involutivity is a natural condition, since any solution of the above open problem using SDP, if it establishes radical ideal membership, will necessarily need (at least implicitly) a real radical Gröbner basis. Our algorithm, uses geometric involutivity, and similarly gives an intermediate ideal, which constitutes another variation on this family of conjectures.

In addition to implementing an algorithm to determine a first facial reduction. We also implemented a test for the existence of additional facial reductions beyond the first (e.g., in the cases of Examples 4.3 and 4.5 of [37]). By using the CVX package or Douglas-Rachford iteration to solve for the auxiliary problem (3.17), we can determine if we need a second facial



reduction by checking whether the optimal value of the auxiliary problem is close to 0. Our implementation of auxiliary facial reductions, as still preliminary and needs improvement. So a more detailed study of this aspect is worthwhile.

Numerical polynomial algebra has been a rapidly expanding and popular area [50]. Its problems are typically very demanding, motivating the implementation of methods to improve accuracy. For example Bertini, the homotopy package developed for numerical polynomial algebra, uses variable precision arithmetic, with particularly demanding problems requiring thousands of digits of precision. Consequently this is also a motivation to develop higher accuracy methods, such as the FDR method of this paper. Manipulations with radical ideals would be a by-product from such work. An important open problem is the following: *Give a numerical algorithm, capable in principle of determining an approximate real witness point on each component of a real variety.* We note that the methods of Wu and Reid [54] and Hauenstein [28] only answer this question under certain conditions, say that the ideal is real radical and defined by a regular sequence. Also see [35], which gives an alternative extension of complex numerical algebraic geometry to the reals, in the complex curve case.

We provided a small set of examples, that illustrate some aspects of our algorithms. In Maple all of our examples were executed with Maple's *Digits* := 15 and the input tolerance :=  $10^{-10}$  for the GIF algorithm which intensively uses LAPack's SVD. Accuracy in the projected residual error for our tests were between  $10^{-14}$  and  $10^{-12}$ . The normalized generators obtained for our experiments had coefficients differing less than  $10^{-10}$  from the exact coefficients.

In addition we prove that our facial reduction steps are backwards stable. See Theorem 3.5.2 and Section 3.5.2. The advantage for the use of Douglas-Rachford iterations in our SDP solution techniques and its linear convergence is discussed at the end of Section 3.5.3. We note that the simplest structured matrices from polynomial systems are Hankel matrices and are notoriously ill-conditioned, see e.g., [8, 25]. In particular such matrices all lie *close* to the boundary of the semidefinite cone. Therefore, even after successful facial reduction guarantees a strictly feasible solution, the set of Hankel matrices are all *nearly singular*. This makes the related feasibility problems particularly difficult. Despite this we were successful in finding feasible solutions. Such conditioning issues warrant further study. Indeed consider  $p(x, y) = x^2 + y^2 + \epsilon = 0$ . Even though  $(x, y) = (0, 0)$  is the unique solution for  $\epsilon = 0$ , with associated real

radical ideal  $\langle x, y \rangle_{\mathbb{R}}$ , the solution is not a real continuous function of  $\epsilon$  as  $\epsilon$  passes through 0. So the problem in terms of the variety is not well-posed. An interesting challenge is to formulate appropriate well-posed nearby problems in an appropriate space. The backwards stable tools, of facial reduction and auxiliary reduction, and associated spaces are interesting possibilities for such approaches.

## Bibliography

- [1] A. Alfakih and H. Wolkowicz. Matrix completion problems. In *Handbook of semidefinite programming*, volume 27 of *Internat. Ser. Oper. Res. Management Sci.*, pages 533–545. Kluwer Acad. Publ., Boston, MA, 2000. 54, 94
- [2] A.F. Anjos and J.B. Lasserre, editors. *Handbook on Semidefinite, Conic and Polynomial Optimization*. International Series in Operations Research & Management Science. Springer-Verlag, 2011. 56, 58, 68
- [3] F.J.A. Artacho, J.M. Borwein, and M.K. Tam. Recent results on Douglas-Rachford methods. *Serdica Mathematical Journal*, 39:313–330, 2013. 73, 105
- [4] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Math.* Springer-Verlag, 2 edition, 2006. 25, 28, 55, 56
- [5] H.H. Bauschke, J.Y. Bello Cruz, T.T.A. Nghia, H.M. Phan, and X. Wang. The rate of linear convergence of the Douglas-Rachford algorithm for subspaces is the cosine of the Friedrichs angle. *J. Approx. Theory*, 185:63–79, 2014. 75, 105
- [6] H.H. Bauschke, P.L. Combettes, and D.R. Luke. Phase retrieval, error reduction algorithm, and Fienup variants: a view from convex optimization. *J. Opt. Soc. Amer. A*, 19(7):1334–1345, 2002. 73
- [7] H.H. Bauschke and D. Noll. On the local convergence of the Douglas-Rachford algorithm. *Arch. Math. (Basel)*, 102(6):589–600, 2014. 75, 105

- [8] B. Beckermann. The condition number of real Vandermonde, Krylov and positive definite Hankel matrices. *Numer. Math.*, 85(4):553–577, 2000. 85
- [9] G. Blekherman, P.A. Parrilo, and R.R. Thomas, editors. *Semidefinite optimization and convex algebraic geometry*, volume 13 of *MOS-SIAM Series on Optimization*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA, 2013. 25, 55, 56, 68, 94
- [10] J. Bochnak, M. Coste, and M.F. Roy. *Real Algebraic Geometry*. *Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge A Series of Modern Surveys in Mathematics*. Springer, 1998. 28, 63
- [11] J. Bonasia, F. Lemaire, G.J. Reid, and L. Zhi. Determination of approximate symmetries of differential equations. *Group Theory and Numerical Analysis*, 39:249, 2005. 24, 31, 33, 35, 59, 61, 112
- [12] J.M. Borwein and M.K. Tam. A Cyclic Douglas–Rachford Iteration Scheme. *J. Optim. Theory Appl.*, 160(1):1–29, 2014. 73, 74, 105
- [13] J.M. Borwein and H. Wolkowicz. Facial reduction for a cone-convex programming problem. *J. Austral. Math. Soc. Ser. A*, 30(3):369–380, 1980/81. 12, 54, 94, 98, 106
- [14] J.M. Borwein and H. Wolkowicz. Regularizing the abstract convex program. *J. Math. Anal. Appl.*, 83(2):495–530, 1981. 12, 54, 94, 98, 106
- [15] Y-L. Cheung, S. Schurr, and H. Wolkowicz. Preprocessing and regularization for degenerate semidefinite programs. In D.H. Bailey, H.H. Bauschke, P. Borwein, F. Garvan, M. Thera, J. Vanderwerff, and H. Wolkowicz, editors, *Computational and Analytical Mathematics, In Honor of Jonathan Borwein’s 60th Birthday*, volume 50 of *Springer Proceedings in Mathematics & Statistics*, pages 225–276. Springer, 2013. 12, 70, 72, 97, 98
- [16] Y.-L. Cheung and H. Wolkowicz. Sensitivity analysis of semidefinite programs without strong duality. Technical report, University of Waterloo, Waterloo, Ontario, 2014. submitted June 2014, 37 pages. 54, 94

- [17] E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91(2, Ser. A):201–213, 2002. 76
- [18] Jim Douglas, JR and H.H. Rachford, JR. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Amer. Math. Soc.*, 82:421–439, 1956. 73, 105
- [19] D. Drusvyatskiy, N. Krislock, Y-L. Cheung Voronin, and H. Wolkowicz. Noisy sensor network localization: robust facial reduction and the Pareto frontier. Technical report, University of Waterloo, Waterloo, Ontario, 2014. arXiv:1410.6852, 20 pages. 12, 54, 94
- [20] D. Drusvyatskiy, G. Li, and H. Wolkowicz. Alternating projections for ill-posed semidefinite feasibility problems. Technical report, University of Waterloo, Waterloo, Ontario, 2014. submitted Sept. 2014, 12 pages. 54, 69, 94, 102
- [21] M. Dür, B. Jargalsaikhan, and G. Still. The Slater condition is generic in linear conic programming. Technical report, University of Trier, Trier, Germany, 2012. 54, 94
- [22] C. Eckart and G. Young. A principal axis transformation for non-Hermitian matrices. *Bull. Amer. Math. Soc.*, 45:118–121, 1939. 72
- [23] A. Edelman and E. Kostlan. How many zeros of a random polynomial are real? *Bull. Amer. Math. Soc. (N.S.)*, 32(1):1–37, 1995. 53, 75
- [24] R. Escalante and M. Raydan. *Alternating projection methods*, volume 8 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2011. 72, 103
- [25] W. Gautschi and G. Inglese. Lower bounds for the condition number of Vandermonde matrices. *Numer. Math.*, 52(3):241–250, 1988. 85
- [26] V.P. Gerdt and Y.A. Blinkov. Involutive bases of polynomial ideals. *Mathematics and Computers in Simulation*, 45(5):519–541, 1998. 7, 24, 29, 59, 112
- [27] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 3<sup>rd</sup> edition, 1996. 71

- [28] Jonathan D Hauenstein. Numerically computing real points on algebraic sets. *Acta applicandae mathematicae*, 125(1):105–119, 2013. 5, 26, 83, 85, 121
- [29] M. Kac. On the average number of real roots of a random algebraic equation. *Bull. Amer. Math. Soc.*, 49:314–320, 1943. 75
- [30] N. Krislock and H. Wolkowicz. Explicit sensor network localization using semidefinite representations and facial reductions. *SIAM Journal on Optimization*, 20(5):2679–2708, 2010. 12, 54, 94
- [31] M. Kuranishi. On e. cartan’s prolongation theorem of exterior differential systems. *American Journal of Mathematics*, pages 1–47, 1957. 24, 29, 33, 59, 112
- [32] J.B. Lasserre, M. Laurent, and P. Rostalski. A prolongation–projection algorithm for computing the finite real variety of an ideal. *Theoretical Computer Science*, 410(27):2685–2700, 2009. 1, 8, 25, 47, 53, 54, 55, 64, 84, 93, 94, 120
- [33] M. Laurent and P. Rostalski. The approach of moments for polynomial equations. In Miguel F. Anjos and Jean B. Lasserre, editors, *Handbook on semidefinite, conic and polynomial optimization*, International Series in Operations Research & Management Science, 166, pages 25–60. Springer, New York, 2012. 10, 13, 14, 15, 16, 17, 36, 38, 40, 46, 47, 58, 95, 96, 128, 129
- [34] P. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979. 73, 105
- [35] Y. Lu, D.J. Bates, A.J. Sommese, and C.W. Wampler. Finding all real points of a complex curve. In *Algebra, geometry and their interactions*, volume 448 of *Contemp. Math.*, pages 183–205. Amer. Math. Soc., Providence, RI, 2007. 85, 121
- [36] Y. Ma. *Polynomial Optimization via Low-rank Matrix Completion and Semidefinite Programming*. PhD thesis, Academy of Mathematics and Systems Science, Chinese Academy of Science, 2012. 8, 53, 55, 62, 77, 84, 93, 120, 121

- [37] Y. Ma, C. Wang, and L. Zhi. A certificate for semidefinite relaxations in computing positive dimensional real varieties. *Journal of Symbolic Computation*, 72:1 – 20, 2016. vii, x, 1, 8, 53, 55, 62, 77, 78, 79, 80, 81, 82, 84, 93, 120, 121, 128, 129
- [38] Y. Ma and L. Zhi. Computing real solutions of polynomial systems via low-rank moment matrix completion. In *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation*, pages 249–256. ACM, 2012. 26, 83
- [39] F.S. Macaulay and P. Roberts. *The algebraic theory of modular systems*. University press Cambridge, 1916. 29, 57
- [40] H.M. Möller and T. Sauer. H-bases for polynomial interpolation and system solving. *Advances in Computational Mathematics*, 12(4):335–362, 2000. 29, 46, 57, 83
- [41] B. Mourrain. Isolated points, duality and residues. *Journal of Pure and Applied Algebra*, 117:469–493, 1997. 29, 46, 57
- [42] B. Mourrain. A new criterion for normal form algorithms. In *Applied algebra, algebraic algorithms and error-correcting codes*, pages 430–442. Springer, 1999. 29, 46, 57
- [43] G.J. Reid, J. Tang, and L. Zhi. A complete symbolic-numeric linear method for camera pose determination. In *Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, pages 215–223. ACM, 2003. 24, 31, 33, 59, 112
- [44] G.J. Reid, F. Wang, and W. Wu. Geometric involutive bases for positive dimensional polynomial ideals and sdp methods. Technical report, Department of Appl. Math., University of Western Ontario, 2014. 57, 59, 60, 62, 112
- [45] G.J. Reid and L. Zhi. Solving polynomial systems via symbolic-numeric reduction to geometric involutive form. *Journal of Symbolic Computation*, 44(3):280–291, 2009. 24, 59, 112
- [46] Jean-Jacques Risler. Une caractérisation des idéaux des variétés algébriques réelles. *CR Acad. Sci. Paris*, 271:1171–1173, 1970. 28, 63

- [47] R. Scott, G.J. Reid, W. Wu, and L. Zhi. Geometric involutive bases and applications to approximate commutative algebra. In Lorenzo Robbiano and John Abbott, editors, *Approximate Commutative Algebra*, pages 99–124. Springer, 2010. 24, 37, 43, 55, 59, 60, 95, 112
- [48] A.J. Sommese and C.W. Wampler. *The Numerical solution of systems of polynomials arising in engineering and science*, volume 99. World Scientific, 2005. 1, 4, 25, 53, 93
- [49] F. Sottile. *Real solutions to equations from geometry*, volume 57 of *University Lecture Series*. American Mathematical Society, Providence, RI, 2011. 8, 53, 54, 55, 56, 84, 93, 94, 120
- [50] Hans J. Stetter. *Numerical polynomial algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2004. 29, 57, 84, 85, 120, 129
- [51] A.D. Wittkopf and G.J. Reid. Fast differential elimination in c: The cdiffelim environment. *Computer Physics Communications*, 139(2):192–217, 2001. 24, 31, 33, 59, 112, 128
- [52] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of semidefinite programming*. International Series in Operations Research & Management Science, 27. Kluwer Academic Publishers, Boston, MA, 2000. Theory, algorithms, and applications. 10, 11, 12, 68
- [53] H. Wolkowicz and Q. Zhao. Semidefinite programming relaxations for the graph partitioning problem. *Discrete Appl. Math.*, 96/97:461–479, 1999. Selected for the special Editors’ Choice, Edition 1999. 54, 94
- [54] W. Wu and G.J. Reid. Finding points on real solution components and applications to differential polynomial systems. In *Proceedings of the 38th international symposium on International symposium on symbolic and algebraic computation*, pages 339–346. ACM, 2013. 5, 26, 83, 85, 121

- [55] X. Wu and L. Zhi. Determining singular solutions of polynomial systems via symbolic–numeric reduction to geometric involutive forms. *Journal of Symbolic Computation*, 47(3):227–238, 2012. 24, 59, 112
- [56] Q. Zhao, S.E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *J. Comb. Optim.*, 2(1):71–109, 1998. Semidefinite programming and interior-point approaches for combinatorial optimization problems (Fields Institute, Toronto, ON, 1996). 54, 94



## Chapter 4

# Maximum Rank Moment Matrices by Facial Reduction and Douglas-Rachford Method

### 4.1 Introduction

The breakthrough work of Lasserre and collaborators [26, 42] shows that the *real radical ideal*,  $RRI$ , of a real polynomial system with finitely many solutions can be determined by maximizing the rank of so-called moment matrices arising from a *semidefinite programming (SDP)* feasibility problem. This RRI is generated by a system of real polynomials having only real roots that are free of multiplicities. The number of such real roots may be considerably less than the number of complex roots (see the paper [34] for examples and references). Global numerical solvers, such as homotopy continuation solvers typically compute all real roots by first computing all complex (including real) roots. And if the roots have multiplicity, then elaborate strategies are needed to avoid difficulties that arise as the paths from the homotopy solvers approach these singular roots [41]. A conjectured extension of such methods to positive dimensional polynomial systems has been given recently by Ma, Wang and Zhi [32, 31]. These extensions depend on the *method of moments* within a SDP formulation.

Our SDP feasibility formulation is a moment problem equivalent to finding a maximum

rank matrix  $X$  for a linear system of the following type

$$\mathcal{A}X = b, \quad X \in \mathcal{S}_+^k, \quad (4.1)$$

where  $\mathcal{S}_+^k$  denotes the convex cone of  $k \times k$  real symmetric positive semidefinite matrices, and  $\mathcal{A} : \mathcal{S}_+^k \rightarrow \mathbb{R}^m$  is a linear transformation. Also see Problem 4.2.1 below.

The standard regularity assumption for (4.1) is the *Slater constraint qualification* or strict feasibility assumption:

$$\text{there exists } X \text{ with } \mathcal{A}X = b, \quad X \in \text{int } \mathcal{S}_+^k. \quad (4.2)$$

We let  $X \geq 0, > 0$  denote  $X \in \mathcal{S}_+^k, \in \text{int } \mathcal{S}_+^k$ , respectively. It is well known that the Slater condition for SDP holds generically, e.g., [19]. Surprisingly, many SDP problems arising from particular applications, and in particular our polynomial system applications, are marginally infeasible, i.e., fail to satisfy strict feasibility. This means that the feasible set lies within the boundary of the cone, and even the slightest perturbation of the data can make the problem infeasible. This creates difficulties with the optimality and duality conditions as well as with numerical algorithms. To help regularize such SDP problems so that *strong duality* holds, facial reduction was introduced in 1982 by Borwein and Wolkowicz [8, 9]. However it was only much later that the power of facial reduction was exhibited in many applications, e.g., [50, 47, 1]. Developing algorithmic implementations of facial reduction that work for large classes of SDP problems and the connections with perturbation and convergence analysis has recently been achieved in e.g., [24, 16, 12, 17].

A polynomial system of maximum degree  $d$  equations in  $n$  variables can be viewed as a linear function of its vector  $x$  of monomials [26, 42]. The vector  $x$  of monomials contains  $N(n, d) = \frac{(d+n)!}{d!n!} = \binom{d+n}{d}$  monomials up to the degree  $d$  of the polynomial system. The convex cone for polynomials are semi-definite moment matrices encoding the real solutions of the polynomial equations and have a generalized Hankel-Macaulay structure which depends only on the number of variables. Remarkable advances have been recently made in this area [26, 42, 5] which is an intersection between optimization and algebraic geometry. In Chapter 3 we established a framework for using facial reduction for such systems and then solving

the regularized smaller SDP problem to determine the RRI. In the approximate case, as in the paper [34] we apply *geometric involutive bases* [38] to the constraints.

In this chapter, we use facial reduction approach to effectively reduce the size of the moment matrix and then solve the reduced problem using the Douglas-Rachford reflection method. Finally we recover the positive semidefinite solution satisfying the constraints with maximum rank. We perform facial reductions on the primal form. We compare the performance of our techniques with the popular SDP solver SeDuMi(CVX) which uses an interior point method. On our illustrative examples, our approach has better accuracy, and the maximum rank condition can be guaranteed without misleading small eigenvalues. We showed that under some assumptions our algorithm is well-posed, i.e., the maximum rank is unchanged under small perturbations and the computed approximate solution converge to the exact solution as the perturbation approaches to zero. We also give a method to compute the generators of real radicals up to a given degree which gives an if and only if condition for checking the real radical membership (checking if a polynomial belongs to the real radical ideal).

## 4.2 Moment Matrices

**Definition 4.2.1 (Moment Matrix [28])** *Given a linear form  $\lambda \in \mathbb{R}[x]^*$ ,  $x = (x_1 \cdots x_n)$  which maps a polynomial to a real number. A symmetric matrix*

$$M(\lambda) = (\lambda(x^\alpha x^\beta))_{\alpha, \beta \in \mathbb{N}^n} \quad (4.3)$$

*is called a moment matrix of  $\lambda$  where  $\mathbb{N} = \{0, 1, 2, \dots\}$ .*

Similarly, we define the truncated moment matrix.

**Definition 4.2.2 (Truncated Moment Matrix [28])** *Given a linear form  $\lambda_d \in (\mathbb{R}[x]_{2d})^*$ , the truncated moment matrix of  $\lambda_d$  is defined to be*

$$M(\lambda_d) = (\lambda_d(x^\alpha x^\beta))_{\alpha, \beta \in \mathbb{N}_d^n} \quad (4.4)$$

*where  $\mathbb{N}_d^n = \{\gamma \in \mathbb{N}^n : |\gamma| = \sum_{j=1}^n \gamma_j \leq d\}$ .*

**Example 4.2.1** Suppose  $\lambda_1 \in \mathbb{R}[x, y]_{2d}^*$  for  $d = 1$ . Then

$$M(\lambda_1) = \begin{bmatrix} u_{00} & u_{10} & u_{01} \\ u_{10} & u_{20} & u_{11} \\ u_{01} & u_{11} & u_{02} \end{bmatrix} \quad (4.5)$$

Without loss of generality, we assume  $u_{00} = 1$  throughout this chapter.

In recent years, the moment matrix has found applications in the field of real algebraic geometry. A very important application is to compute the real radical ideal of a polynomial system by computing its kernel provided the rank of the moment matrix is maximum and the moment matrix is positive-semidefinite [28]. Existing methods to compute such maximum rank matrices are not accurate. So our main problem in this chapter is the following:

**Problem 4.2.1 (Primal Form Feasibility Problem)** Given  $\mathcal{A} : \mathcal{S}^k \rightarrow \mathbb{R}^m$  as a linear transformation,  $B \in \mathbb{R}^{k \times l}$ , our main problem is the following feasibility problem for the moment matrix  $M$ :

$$\text{Find a max } (\text{rank } M) \in \mathcal{S}_+^k : \mathcal{A}(M) = b, B^T M = 0. \quad (4.6)$$

Here  $\mathcal{A}(M) = \left( \langle A_t, M \rangle \right)_{t \in \mathcal{E}} \in \mathbb{R}^m$ ,  $\mathcal{E} = \{1, 2, \dots, r\}$  and  $r$  is the total number of the linear constraints of  $\mathcal{A}$ , the inner product is trace inner product. The full column rank matrix  $B$  is the coefficient matrix of a polynomial system. In this chapter we are particularly interested in the case of real polynomials; and  $A_t$  can be derived for our application by Algorithm 5.1 described in [34] such that  $\mathcal{A}(M) = b$  enforces the moment matrix structure of  $M$  defined in (4.2.2).  $b \in \mathbb{R}^m$  such that the first entry of  $b$  is 1 and the others are zero. For more details, see [34].

### 4.3 SDP and facial reduction

Consider the semidefinite programming *primal feasibility problem* in its standard form:

$$F_P := \{X : \mathcal{A}(X) = b, X \in \mathcal{S}_+^k\}, \quad (4.7)$$

where  $\mathcal{S}_+^k$  denotes the convex cone of  $k \times k$  real symmetric positive semidefinite matrices,  $\mathcal{A} : \mathcal{S}^k \rightarrow \mathbb{R}^m$  is a linear transformation and  $\mathcal{S}^k$  denotes  $k \times k$  symmetric real matrices.

The semidefinite *dual feasibility problem* is

$$F_D := \{Z : Z = C - \mathcal{A}^*y, Z \in \mathcal{S}_+^k\}, \quad (4.8)$$

where  $\mathcal{A}^*$  is the adjoint of  $\mathcal{A}$  defined as  $\mathcal{A}^*y = \sum_{i=1}^m y_i A_i$  and  $C$  is a constant matrix.

The linear transform  $\mathcal{A}$  can be represented as a matrix form such that  $\mathcal{A}(X) = A \cdot \text{s2vec}(X)$  where  $\text{s2vec}(X)$  is the vectorization of  $X$ . We denote  $A$  as the matrix form of  $\mathcal{A}$ . When we say  $\mathcal{A}$  is linearly independent, we mean  $A$  has linearly independent rows.

### 4.3.1 Faces

**Definition 4.3.1** Given convex cones  $F, K$  and  $F \subseteq K$ , we call  $F$  a face,  $F \trianglelefteq K$  if

$$x, y \in K, x + y \in F \implies x, y \in F.$$

The conjugate face of  $F \trianglelefteq K$ ,  $F^c$  is

$$F^c = F^\perp \cap K.$$

Given a nonempty convex subset  $S$  of  $K$ , the minimal face of  $K$  containing  $S$ , denoted as  $\text{face}(S, K)$ , is defined to be the intersection of all faces of  $K$  containing  $S$ .

The following properties of minimal face in the convex cone  $\mathcal{S}_+^n$  are well known [11].

**Proposition 4.3.1** Let  $X \in \mathcal{S}_+^n$  have rank  $r$  and let

$$X = \begin{bmatrix} P & Q \end{bmatrix} \begin{bmatrix} D_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P & Q \end{bmatrix}^T, \quad D_r \in \mathcal{S}_{++}^r$$

be its spectral decomposition and  $\mathcal{S}_{++}^r$  denotes the convex cone of  $r \times r$  real symmetric positive definite matrices. Then the minimal face,  $\text{face}(X, \mathcal{S}_+^n)$ , and its conjugate face satisfy

$$\text{face}(X, \mathcal{S}_+^n) = P \mathcal{S}_+^r P^T, \quad \text{face}(X, \mathcal{S}_+^n)^c = Q \mathcal{S}_+^{n-r} Q^T.$$

### 4.3.2 Theorems of the alternative

The following two theorems introduce key concepts for facial reduction.

**Theorem 4.3.1 (Primal Theorem of alternative [11, 18])** Suppose  $\mathcal{A} : \mathcal{S}_+^k \rightarrow \mathbb{R}^m$  is a linear transformation,  $b \in \mathbb{R}^m$ ,  $P \in \mathcal{S}^k$  and  $Z \in \mathcal{S}^k$ . Then exactly one of the following alternative systems is consistent:

$$(I) \quad 0 < P \in F := \{P \in \mathcal{S}^k : \mathcal{A}(P) = b, P \geq 0\} \quad (\text{Slater}) \quad (4.9a)$$

$$(II) \quad 0 \neq Z \in D := \{Z \in \mathcal{S}^k : Z = \mathcal{A}^*y \geq 0, b^T y = 0\}. \quad (\text{Auxiliary}) \quad (4.9b)$$

**Proof** Note that if (II) is consistent, then  $Z$  exposes a face of  $\mathcal{S}_+^n$  that contains the minimal face  $(F, \mathcal{S}_+^n)$ . That is, for  $P \in F$  we have

$$\text{trace } ZP = \text{trace}(\mathcal{A}^*y)P = y^T b = 0.$$

The remainder of the proof can be found in [11, 18] or Appendix A.

Equation (4.9a) is called the *primal problem* and equation (4.9b) is called the *auxiliary problem*.

The theorem of alternative for the dual form follows.

**Theorem 4.3.2 ( Theorem of alternative for dual form [8, 9])** Suppose  $\mathcal{A} : \mathcal{S}_+^k \rightarrow \mathbb{R}^m$  is a linear transformation,  $P \in \mathcal{S}^k, Z \in \mathcal{S}^k$ . Then exactly one of the following alternative systems is consistent:

$$(I) \quad Z = C - \mathcal{A}^*y > 0 \quad (4.10a)$$

$$(II) \quad \mathcal{A}(X) = 0, \langle C, X \rangle = 0, X \geq 0 \implies X = 0. \quad (4.10b)$$

See [11] for a detailed proof.

### 4.3.3 Facial reduction

Recall Theorem 4.3.1, that when (4.9a) is true, the *Slater condition* holds. The Slater condition is an important concept in optimization. The failure of the Slater condition usually results in poor performance of algorithms such as interior point methods and the Douglas-Rachford method. Facial reduction aims to regularize an SDP problem so that the Slater condition holds on a minimal face.

**Lemma 4.3.3 (Facial reduction on the primal form)** *Suppose  $F_P$  is non-empty. Then*

$$\left\{ \begin{array}{l} \mathcal{A}(P) = b, P \in \mathcal{S}_+^k \\ 0 \neq Z = \mathcal{A}^*y \in \mathcal{S}_+^k, b^T y = 0 \end{array} \right\} \Rightarrow P \in \{Z\}^\perp \cap \mathcal{S}_+^k \quad (4.11)$$

Such a  $Z$  is called an *exposing vector* of  $\mathcal{S}_+^k$ . By solving the second problem in the bracket, we can get an exposing vector which reduces the primal problem (4.9a) to a smaller face, i.e.,  $\{Z\}^\perp \cap \mathcal{S}_+^k$  which is reformulated as a primal feasibility problem on a smaller cone  $\mathcal{S}_+^{\bar{k}}, \bar{k} < k$  described in Theorem 4.3.4. The process is repeated until we get face  $(F_P, \mathcal{S}_+^k)$ , the minimal face of  $\mathcal{S}_+^k$  containing  $F_P$ , and the Slater condition (4.9a) holds.

Based on the two statements of the theorems of alternative, we can always apply facial reduction to the dual or primal form to reduce the dimension of the problem. In this chapter, we express our moment matrix problem in the primal form yielding greater accuracy in our examples when solved using facial reductions and the Douglas-Rachford (DR) method. Details of DR are given later in Section 4.4.

Suppose an exposing vector is found. The following theorems shows how to use the exposing vector to get an equivalent problem on a smaller positive semidefinite cone so that an additional facial reduction can be done.

**Theorem 4.3.4** *Suppose  $\mathcal{A} : \mathcal{S}^k \rightarrow \mathbb{R}^m$  is a linear transformation as in Problem 4.2.1,  $P \in \mathcal{S}^k$ ,  $Z \in \mathcal{S}_+^k$  is an exposing vector,  $Z = \begin{bmatrix} U & V \end{bmatrix} \begin{bmatrix} D_l & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U & V \end{bmatrix}^T$  is the spectral decomposition. Suppose  $\bar{A}_l := V^T A_l V$  and  $\bar{\mathcal{A}} : \mathcal{S}^d \rightarrow \mathbb{R}^m$  is the linear transformation induced by  $\bar{A}_l$  where  $d + l = k$ . Then*

$$\exists P \in \mathcal{S}^k, \mathcal{A}(P) = b, Z^T P = 0, P \geq 0 \quad (4.12a)$$

$$\iff$$

$$\exists \bar{P} \in \mathcal{S}^d, \bar{\mathcal{A}}(\bar{P}) = \bar{b}, \bar{P} \geq 0. \quad (4.12b)$$

**Proof** First, we assume  $\bar{b} = b$ .

To show (4.12a) implies (4.12b). Suppose there exists  $P \geq 0$  satisfying (4.12a). Apply the spectral decomposition to  $P$ . Then we have  $P = U_1 P_1 U_1^T$  where  $U_1^T U_1 = I, Z^T U_1 = 0$ ,  $P_1 > 0$  (choosing only the positive eigenvalues) and  $\text{rank}(U_1) \leq \text{rank}(V)$ . Let  $Q$  be a linear

transformation such that  $VQ = U_1$ . Then  $\text{trace}(A_t V Q P_1 Q^T V^T) = \text{trace}(A_t U_1 P_1 U_1^T) = \mathcal{A}(P) = b$ . Hence we conclude  $\exists \bar{P} = Q P_1 Q^T, \bar{\mathcal{A}}(\bar{P}) = b, \bar{P} \succeq 0$ .

To show (4.12b) implies (4.12a), note that the existence of  $\bar{P}$  satisfying (4.12b) implies that  $P = V \bar{P} V^T$  satisfies (4.12a).

We assume  $\mathcal{A}$  is linearly independent, however,  $\bar{\mathcal{A}}$  is not necessarily linearly independent. We can remove the redundant linear constraints in  $\bar{\mathcal{A}}$  and the corresponding elements in  $b$  to obtain  $\bar{b}$ . So without loss of generality, we have  $\bar{\mathcal{A}}(\bar{P}) = \bar{b}, \bar{P} \succeq 0, \bar{\mathcal{A}}$  is linearly independent. ■

### 4.3.4 Facial reduction maximum rank algorithm

Recall from the Primal Form Feasibility Problem 4.2.1, we can just set  $Z = BB^T$  as the exposing vector to do the first facial reduction as described in Theorem 4.3.4.

To do more facial reductions, after the first facial reduction, the problem is considered in the form of (4.9a). Then according to Theorem 4.3.1, we need to determine if (4.9a) is strictly feasible, i.e. to determine if there exists a  $P \succ 0$ . We need to solve the following auxiliary problem:<sup>1</sup>

$$\begin{aligned} p^*(\mathcal{A}, b) := \min_y \quad & \frac{1}{2}(b^T y)^2 \\ \text{s.t.} \quad & Z = \mathcal{A}^* y \succeq 0 \\ & \text{trace } \mathcal{A}^* y = 1. \end{aligned} \tag{4.13}$$

We set  $\text{trace } \mathcal{A}^* y = 1$  because we need to rule out  $y$  being the zero solution. If we solve this problem successfully with  $|b^T y| = 0$  with a non-zero  $y$ , we have  $Z = \mathcal{A}^* y \neq 0$ . By Theorem 4.3.1, (4.9a) only admits a positive semidefinite but no positive definite solution which indicates Slater condition fails and a second facial reduction is needed. We then use this  $Z$  as the exposing vector to do the second facial reduction as described in Theorem 4.3.4. We repeat this process until  $p^*(\mathcal{A}, b)$  is strictly positive which means there exists a positive definite solution of (4.12b) and that the Slater condition holds.

The algorithm to use facial reduction to find maximum rank solutions is summarized as

---

<sup>1</sup> This can be implemented in e.g., CVX using the *norm* function or absolute value function for the objective, i.e., we minimize  $|\bar{b}^T y|$  rather than using the squared term.



follows:

---

**Algorithm 4.3.1:** Facial reduction on the primal.

---

```

1 Input( $\mathcal{A} : \mathcal{S}^n \rightarrow \mathbb{R}^m, b \in \mathbb{R}^m, B \in \mathbb{R}^{k \times l}$  as in Problem 4.2.1, set  $p^*(\mathcal{A}, b) = 0, W = I$ );
2 repeat
3   Find the exposing vector  $Z$  by setting  $Z = BB^T$  (first facial reduction) or solving the
   auxiliary problem (4.13) for  $p^*(\mathcal{A}, b)$ .
4   if  $p^*(\mathcal{A}, b) > 0$  then
5     STOP, facial reduction finished, Slater condition holds
6   else
7     Apply eigenvalue decomposition to  $Z$  to obtain  $V$  such that  $V$  is the nullspace of
      $Z$  and  $V^T V = I$ .
8     Update  $\mathcal{A}$  such that  $A_i \leftarrow V^T A_i V, \forall i \in \mathcal{E}$ , then update  $\mathcal{A}, b$  by removing
     redundant relations.
9     Update  $W$  by  $W \leftarrow W \cdot V$ .
10  end if
11 until  $p^*(\mathcal{A}, b) > 0$ ;
12 Solve  $\mathcal{A}(P) = b, P \succ 0$ . Recover the moment matrix  $M = WPW^T$ .
13 Output( $M$  which is maximum rank)

```

---

**Theorem 4.3.5 (Maximum rank)** *No further facial reductions can be done if and only if  $p^*(\mathcal{A}, b) > 0$ . Algorithm 4.3.1 returns a maximum rank solution of Problem 4.2.1.*

**Proof** After the first facial reduction, Problem 4.2.1 is transformed into (4.12a). By the proof of Theorem 4.3.1, each time when we find a solution  $Z \geq 0$  from (4.9b), we can find the feasible solutions of (4.9a) lie in the nullspace of  $Z$ . By Theorem 4.3.4, we can reduce the problem to an equivalent smaller problem without loss of information. By Theorem 4.3.1 when we have  $p^*(\mathcal{A}, b) > 0$ , we have reduced the problem to a minimal face where (4.9a) admits a positive semidefinite solution, which is equivalent to saying no further facial reductions can be done. So if  $p^*(\mathcal{A}, b) > 0$ , there exists  $P \succ 0$  such that  $\mathcal{A}(P) = b$ . Since the minimal face contains all the feasible solutions of Problem 4.2.1 and  $P$  is the maximum rank solution on the minimal face,  $M$  is also the maximum rank solution of Problem 4.2.1. ■

**Singularity degree** The minimal number of facial reduction steps is called *singularity degree*. The examples in Section 4.10 show that some examples with singularity more than 1 can be accurately solved by Facial reduction heuristics. For more details, see [45, 17].

### 4.3.5 Transform of the auxiliary problem

The auxiliary problem (4.13) can be solved by CVX or other SDP solvers, but in order to get higher accuracy, we use Douglas-Rachford iteration. To do that, we need to reformulate the auxiliary problem.

**Definition 4.3.1** Given a matrix  $A = (a_{ij})_{1 \leq i, j \leq n} \in \mathcal{S}_+^n$ , define  $\text{vec}(A)$  to be the vectorization of  $A$ , i.e.,

$$\text{vec}(A) = [a_{11}, a_{12}, \dots, a_{1n}, a_{21}, a_{22}, \dots, a_{n1}, \dots, a_{nn}]^T$$

Suppose  $A$  is the matrix form of  $\mathcal{A}$ , i.e.,  $A = [\text{vec}(A_1), \dots, \text{vec}(A_m)]^T$ , then problem (4.13) can be converted to:

$$\begin{aligned} \text{Find } y \in \mathbb{R}^m : b^T y &= 0, A^T y - \text{vec}(Z) = 0, \\ Z &\geq 0, \text{trace}(Z) = 1. \end{aligned} \tag{4.14}$$

Problem (4.14) is equivalent to

$$K \cdot W = R, Z \geq 0, \tag{4.15}$$

where  $K = [b^T, 0^T; A^T, -I; 0, \text{vec}(I)]$ ,  $W = (y; \text{vec}(Z))$  and  $R = [0; 0; 1]$ .

In addition, we could lower the dimension of  $W$  using the following theorem:

**Theorem 4.3.6** Suppose  $K, W, R, A^T$  are defined as above in (4.15),  $(A^T)^\dagger$  is the Moore-Penrose pseudoinverse of  $A^T$ . Suppose  $L = [b^T \cdot (A^T)^\dagger; I - A^T \cdot (A^T)^\dagger; \text{vec}(I)]$  and  $R = [0; 0; 1]$ . Then

$$K \cdot W = R, Z \geq 0, \tag{4.16}$$

$$\Longleftrightarrow$$

$$L \cdot \text{vec}(Z) = R, Z \geq 0, \tag{4.17}$$

**Proof** Let's assume  $\text{vec}(Z) = A^T y$ , then we have  $A^T(A^T)^\dagger \text{vec}(Z) = A^T(A^T)^\dagger A^T y = A^T y = \text{vec}(Z)$  since  $(A^T)^\dagger A^T = I$ . Also  $(A^T)^\dagger \text{vec}(Z) = (A^T)^\dagger A^T y = y$ .

It is easy to verify the other direction, by making the substitution  $y = (A^T)^\dagger \text{vec}(Z)$ . ■

By our experiments, we found this formulation has the best performance when coupled with Douglas-Rachford methods. So we use (4.17) to solve the auxiliary problem (4.9b).

## 4.4 Projection method

In Algorithm 4.3.1, we need to solve two problems: the auxiliary problem to solve is (4.17) and the primal problem after facial reduction to solve is  $\mathcal{A}(P) = b, P \succ 0$ . Essentially, we need to find the intersection between an affine subspace (linear constraints) and a positive semidefinite cone. We consider the *Douglas-Rachford reflection-projection* (DR) method which involves projections and reflections between two convex sets. These two convex sets are the affine subspace and the positive semidefinite cone in our case. There are also other projection-based methods, such as method of *alternating projection* [20]. We prefer the DR method as it displays better convergence properties in our tests. Also, unlike the alternating projection method, which is likely to converge to the boundary of cone, the DR method is likely to converge to the interior of the cone, since we need to solve  $\mathcal{A}(P) = b, P \succ 0$ .

### 4.4.1 Projection to the positive semidefinite cone

Given  $X \in \mathcal{S}^k$ , denote  $\mathcal{P}_{\mathcal{S}_+^k}(X, r)$  as the projection of  $X$  to  $\mathcal{S}_+^k$  such that the projected matrix has rank  $r$ , we have the following theorem:

**Theorem 4.4.1** [23] Suppose  $X \in \mathcal{S}^k$ , the projection of  $\mathcal{P}_{\mathcal{S}_+^k}(X, r)$  with  $r \leq d$  is:  $\mathcal{P}_{\mathcal{S}_+^k}(X, r) = V\mathcal{P}_{\mathcal{S}_+^k}(D, r)V^T$  and  $X = VDV^T$  is the eigenvalue decomposition of  $X$  and  $D$  is a diagonal matrix with all the eigenvalues of  $X$ .  $\mathcal{P}_{\mathcal{S}_+^k}(D, r)$  is obtained by keeping the first  $r$  largest positive eigenvalues unchanged while setting all the other eigenvalues to zero.

### 4.4.2 Projection to an affine subspace

Suppose an affine subspace is given as follows:

$$\{X \in \mathcal{S}^k, \mathcal{A}(X) = b\} \quad (4.18)$$

or equivalently

$$\{X \in \mathcal{S}^k, A \cdot \text{s2vec}(X) = b\} \quad (4.19)$$

where  $A$  is the the matrix form of  $\mathcal{A}$  and  $\text{s2vec}(X)$  is the vectorization of  $X$ . To project  $X$  from  $\mathcal{S}^k$  onto the affine subspace (4.19), we have the following well-known theorem:

**Theorem 4.4.2** [33] *Given  $\bar{X} \in \mathcal{S}^k$ ,  $A, b$  defined in (4.19), assume the rows of  $A$  are linearly independent. Let  $A^\dagger$  be the Moore-Penrose pseudoinverse of  $A$  so  $A^\dagger = A^T(AA^T)^{-1}$ .*

$$\begin{aligned} \text{Suppose } X^* &:= \operatorname{argmin}\{\|X - \bar{X}\| : AX = b\} \\ \text{Then } X^* &= \bar{X} + A^\dagger(b - A\bar{X}). \end{aligned} \quad (4.20)$$

We denote  $X^* = \mathcal{P}_{\mathcal{A}}(X)$ .

**Proof** Denote  $R = b - A\bar{X}$ . First, we need to check  $X^* = \bar{X} + A^\dagger R$  is on the linear subspace. So  $AX^* = A(\bar{X} + A^\dagger R) = A\bar{X} + AA^\dagger R = A\bar{X} + b - A\bar{X} = b$ .

We also need to check  $X^*$  is the optimal one. Suppose  $Y^* = \bar{X} + Y$  is on the linear subspace, so  $AY^* = b$ , and  $A(Y^* - X^*) = A(Y - A^\dagger R) = 0$ .

Then  $(Y - A^\dagger R)^T A^\dagger R = (Y - A^\dagger R)^T A^T (AA^T)^{-1} R = (A(Y - A^\dagger R))^T (AA^T)^{-1} R = 0$ . Hence  $\|Y^* - \bar{X}\|^2 = \|Y\|^2 = \|Y - A^\dagger R + A^\dagger R\|^2 = \|Y - A^\dagger R\|^2 + \|A^\dagger R\|^2 \geq \|A^\dagger R\|^2 = \|X^* - \bar{X}\|^2$ .

### 4.4.3 Douglas-Rachford method

In Sections 4.4.1 and 4.4.2 we showed how to project a matrix to a positive semidefinite cone and a affine subspace. Briefly speaking, the DR methods first project a matrix  $X$  to the positive semidefinite cone, then reflect it by multiplying the projected matrix by 2 and subtracting  $X$  from it. Similarly, the resulting matrix is projected and reflected over an affine subspace as well. Finally the average of the original matrix and the reflected matrix is taken to update  $X$  to  $X_{new}$ .

The convergence rate of DR method is studied by Bauschke et al [3, 4]. The original idea about the Douglas-Rachford method came from solving partial differential equations [15]. Then later Lions and Mercier brought the Douglas-Rachford method to light using by connecting it to convex analysis [29]. (More details about the DR method can be found in e.g., [2, 7].) We apply Douglas-Rachford to solve both the primal problem and the auxiliary problem. One step of Douglas-Rachford method is the following:

$$\begin{aligned} Y &= 2\mathcal{P}_{\mathcal{S}_+^k}(X, r) - X, \\ Z &= 2\mathcal{P}_{\mathcal{A}}(Y) - Y, \\ X_{new} &= (X + Z)/2. \end{aligned} \tag{4.21}$$

At each step, we calculate the residual  $Res := \|\mathcal{A}(Y) - b\|$ , which is the residual after projecting onto the positive semidefinite cone. If the residual is less than the given tolerance, we stop and return  $Y$ . According to the basic theorem on the convergence of the sequence, [7, Thm 3.3, Page 11], the residuals of the projections of the iterates on one of the sets have to be used for the stopping criteria. We use the residual after the projection onto the SDP cone since we want our final matrix to be positive semidefinite.

## 4.5 The ill-conditioned case

In practice, some problems appear to be very ill-conditioned. One example is the geometric polynomial in Section 4.10. Those examples have eigenvalue decomposition of the solutions from the auxiliary problem with some eigenvalues that are very small compared to the others, and the DR iterations converge very slowly. This indicates the rank  $r$  used in the projection  $\mathcal{P}_{\mathcal{S}_+^k}(X, r)$  can not be maximum.

To deal with such problems, we would have to project the matrix to a good rank  $r$  matrix using  $\mathcal{P}_{\mathcal{S}_+^k}(X, r)$  as described in Theorem 4.4.1 when solving the auxiliary problem (4.9b) or (4.10b). In other words, at each step of facial reduction, we are not computing the smallest possible face. Instead, we try to find a bigger but much more accurate face. So we may need more facial reductions but we can obtain more accurate results.

The strategy we used to get this good matrix is to look at the eigenvalues of  $Z$  in (4.9b). We drop the eigenvalues which are significantly smaller than the other eigenvalues and  $r$  is chosen

to be the number of eigenvalues which are well conditioned. For example, if the eigenvalues are 0.7, 0.2, 0.00002, 0, 0, 0, we will set  $r = 2$  instead of 3 or 6. After this, we will resolve (4.9b) with the updated  $r$  and  $\mathcal{P}_{\mathcal{S}_+^k}(X, r)$  to obtain a more accurate face.

## 4.6 Well-posedness

In this section, we study the well-posedness of our facial reduction maximum rank Algorithm 4.3.1. We want to show for sufficiently small perturbations of the input, the rank of the minimal face doesn't change. Also, as the perturbation converges to zero, the solution itself converges to the exact solution. The existence of the exact solution of maximum rank is due to Borwein and Wolkowicz [8, 9].

First we introduce the following theorem about the continuity of the Moore-Penrose Pseudoinverse by G.W. Stewart.

**Theorem 4.6.1** [44] *Suppose  $A$  is a matrix. Then*

$$\lim_{\delta A \rightarrow 0} (A + \delta A)^\dagger = A^\dagger \text{ if and only if } \exists \epsilon > 0 : \text{rank}(A + \delta A) = \text{rank}(A) \text{ for all } \delta A : \|\delta A\| < \epsilon. \quad (4.22)$$

Next we introduce a theorem about the perturbation of the primal SDP problem from [40].

**Theorem 4.6.2** [40] *Suppose  $A = [\text{vec}(A_1), \dots, \text{vec}(A_m)]^T$ . Let  $\tilde{A} = A + \delta A$  and  $\text{rank } \tilde{A} = \text{rank } A$ . Suppose  $A \cdot \text{vec}(X) = b, X \succ 0$ , then there exists  $\tilde{X}$  such that  $\tilde{A} \cdot \text{vec}(\tilde{X}) = b + \delta b, \tilde{X} \succ 0$  for sufficiently small  $\delta A$  and  $\delta b$ .*

**Proof** Suppose  $X \succ 0, A \cdot \text{vec}(X) = b$ . Let  $\text{vec}(\tilde{X}) = \text{vec}(X) - \tilde{A}^\dagger \tilde{A} \text{vec}(X) + \tilde{A}^\dagger (b + \delta b)$  where  $\tilde{A}^\dagger$  is the Moore-Penrose pseudoinverse of  $\tilde{A}$ . Then  $\tilde{A} \text{vec}(\tilde{X}) = \tilde{A}(\text{vec}(X) - \tilde{A}^\dagger \tilde{A} \text{vec}(X) + \tilde{A}^\dagger (b + \delta b)) = \tilde{A} \text{vec}(X) - \tilde{A} \text{vec}(X) + b + \delta b = b + \delta b$ .

Now  $\|\tilde{X} - X\| = \|\tilde{A}^\dagger \tilde{A} \text{vec}(X) - \tilde{A}^\dagger b - \tilde{A}^\dagger \delta b\| = \|\tilde{A}^\dagger \tilde{A} \text{vec}(X) - \tilde{A}^\dagger \tilde{A} \text{vec}(X) - \tilde{A}^\dagger \delta b\| = \|\tilde{A}^\dagger (\tilde{A} - A) \text{vec}(X) - \tilde{A}^\dagger \delta b\| \leq \|\tilde{A}^\dagger\| \|\tilde{A} - A\| \|X\| + \|\tilde{A}^\dagger\| \|\delta b\|$ . Since  $A$  is linearly independent, we have  $\text{rank}(\tilde{A}) = \text{rank}(A)$  for small  $\delta A$ , which means  $\tilde{A}^\dagger \rightarrow A^\dagger$  as  $\tilde{A} \rightarrow A$  by Theorem 4.6.1. Therefore  $\|\tilde{X} - X\| \rightarrow 0$  as  $\tilde{A} \rightarrow A$  and  $\delta b \rightarrow 0$ . Since  $X$  is in the interior of the cone  $\mathcal{S}_+^n$ ,  $\tilde{X}$  is also in the interior of  $\mathcal{S}_+^n$  if  $\tilde{X}$  is close enough to  $X$ .

**Theorem 4.6.3** *Assume  $\text{rank}(A + \delta A) = \text{rank } A$ , then for sufficiently small enough  $(\delta A, \delta b)$ , the existence of  $\tilde{X}$  such that  $(A + \delta A) \cdot \text{vec}(\tilde{X}) = b + \delta b, \tilde{X} > 0$  implies that there exists  $X$  such that  $A \cdot \text{vec}(X) = b, X > 0$  for small  $\delta A, \delta b$*

**Proof** Consequence of Theorem 4.6.1.

**Theorem 4.6.4** *Suppose  $\tilde{A} \rightarrow A, \text{rank } \tilde{A} = \text{rank } A$ ,  $L$  is defined as 4.17, then  $\tilde{L} \rightarrow L$ .*

**Proof** Since  $L = [b^T \cdot (A^T)^\dagger; I - A^T \cdot (A^T)^\dagger; \text{vec}(I)]$ , we have  $\tilde{L} = [b^T \cdot (\tilde{A}^T)^\dagger; I - \tilde{A}^T \cdot (\tilde{A}^T)^\dagger; \text{vec}(I)]$ . By Theorem 4.6.1,  $\tilde{A}^\dagger \rightarrow A^\dagger$ , so we have  $\tilde{L} \rightarrow L$ .

**Theorem 4.6.5** *Denote  $L$  from 4.17 as  $L = [\text{vec}(L_1), \dots, \text{vec}(L_m)]^T$  and  $\tilde{L} = L + \delta L = [\text{vec}(\tilde{L}_1), \dots, \text{vec}(\tilde{L}_m)]^T$ . Suppose  $\tilde{L} \cdot \text{vec}(\tilde{X}) = E + \delta E, \tilde{X} \geq 0, \tilde{X} = UPU^T$  where  $P > 0$  is an  $r \times r$  diagonal matrix. Denote  $H = [\text{vec}(U^T L_1 U), \dots, \text{vec}(U^T L_m U)]^T$  and  $\tilde{H} = [\text{vec}(U^T \tilde{L}_1 U), \dots, \text{vec}(U^T \tilde{L}_m U)]^T$ . Assume in addition  $\text{rank } H = \text{rank } \tilde{H}$ , then there exists  $X \geq 0, L \cdot \text{vec}(X) = E$  such that  $X \rightarrow \tilde{X}$  as  $\delta E, \delta L \rightarrow 0$  and  $\text{rank } \tilde{X} = \text{rank } X = r$  for sufficiently small  $\delta L, \delta E$ .*

**Proof** First, one can verify that  $\tilde{H} \cdot \text{vec}(P) = E + \delta E$ . We need to prove that there exists  $\bar{P}$  such that  $H \cdot \text{vec}(\bar{P}) = E, \bar{P} > 0$  for sufficiently small  $\delta E, \delta L$ .

Since  $\tilde{L} \rightarrow L$ , we have  $\tilde{H} \rightarrow H$ . Also  $\text{rank } H = \text{rank } \tilde{H}$  by assumption, according to Theorem 4.6.3, there exists  $\bar{P}$  such that  $H \cdot \text{vec}(\bar{P}) = E, \bar{P} > 0$  for sufficiently small  $\delta E, \delta L$ . Now let  $X = U\bar{P}U^T$ , then  $X \geq 0, L \cdot \text{vec}(X) = E$  such that  $X \rightarrow \tilde{X}$  as  $\delta E, \delta L \rightarrow 0$  and  $\text{rank } \tilde{X} = \text{rank } X = r$  for sufficiently small  $\delta E, \delta L$ . ■

Now, recall Algorithm 4.3.1 for doing facial reductions. At each step, we solve the auxiliary problem (4.14) which is equivalent to solving (4.17) to obtain an exposing vector  $Z$ . Then we compute  $Q = \text{null}(Z)$  to do the next step of facial reduction. That is we update  $A$  by setting  $A \leftarrow [\text{vec}(Q^T A_1 Q), \dots, \text{vec}(Q^T A_m Q)]^T$ . At the end of the algorithm, we obtain a sequence of exposing vectors  $Z^{(1)}, Z^{(2)}, \dots$  and decreasing faces  $Q^{(1)}, Q^{(2)}, \dots$ . Due to numerical error, the

auxiliary problem (4.17) can not be solved exactly. Instead we solved an approximate problem with a small residual  $\delta R$ , that is we solved  $\tilde{L} \cdot \text{vec}(Z) = R + \delta R, Z \succeq 0$  exactly. By Theorem 4.6.5, there exists  $\bar{Z}$  such that  $L \cdot \text{vec}(\bar{Z}) = R, \bar{Z} \succeq 0, \text{rank}(Z) = \text{rank}(\bar{Z})$  and  $\bar{Z} \rightarrow Z$  as  $\delta R, \delta L \rightarrow 0$ . The assumption of Theorem 4.6.5 is satisfied if  $\tilde{H}$  has the full column rank and the singular values are greater than a threshold (much larger than the residual) since  $H$  and  $\tilde{H}$  has more rows than columns. So the approximate face  $Q = \text{null}(Z)$  converges to the exact face  $\bar{Q} = \text{null}(\bar{Z})$  and  $\text{rank } Q = \text{rank } \bar{Q}$ . This step can be repeated so the approximate minimal face  $Q_{\min}$  converges the exact minimal face  $\bar{Q}_{\min}$  and  $\text{rank } Q_{\min} = \text{rank } \bar{Q}_{\min}$  for small perturbations. Finally, by Theorem 4.6.3, the maximum rank of the solutions doesn't change under sufficiently small perturbations and the approximation solution converges to the exact solutions if the perturbation converges to 0.

So we have proved the following well-posedness theorem.

**Theorem 4.6.6** *The maximum rank of the output from Algorithm 4.3.1 doesn't change if the residual at each facial reduction is small enough. The output approximate matrix from Algorithm 4.3.1 converges to the exact solution if the residual at each facial reduction converges to zero.*

We also direct the readers to the very interesting related work [40] where well-posedness is considered under stronger assumptions.

## 4.7 Computation of generators of the real radical up to a given degree

Based on the maximum rank moment matrix, the geometric involutive form [34], the results of Curto and Fialkow [14] and Lasserre et al. [27] we give an algorithm for computing the real radical up to a given degree  $d$ .

Throughout this section we consider a system of multivariate polynomials  $\{f_1, \dots, f_m\} \subseteq \mathbb{R}[x_1, x_2, \dots, x_n]$  of degree  $d = \max_i(\deg(f_i))$ . The associated real ideal is denoted

$$I := \langle f_1, f_2, \dots, f_m \rangle_{\mathbb{R}} \quad (4.23)$$



and its associated real radical ideal is denoted by  $\sqrt[\mathbb{R}]{I}$ .

In particular we solve the following problem:

**Problem 4.7.1** *Given a system of polynomials  $\{f_1, \dots, f_m\} \subseteq \mathbb{R}[x_1, x_2, \dots, x_n]$  with associated ideal  $I$  and an integer  $d$  we give an algorithm to compute:*

$$\left(\sqrt[\mathbb{R}]{I}\right)_{(\leq d)} := \{f \in \sqrt[\mathbb{R}]{I} : \deg(f) \leq d\} \quad (4.24)$$

We will represent  $\left(\sqrt[\mathbb{R}]{I}\right)_{(\leq d)}$  by polynomials corresponding to vectors in  $\ker M(\lambda_d)$  where  $M(\lambda_d)$  is the truncated moment matrix to degree  $d$  as defined in Definition 4.2.2.

In order to obtain our main result we will require that  $\ker M(\lambda_d)$  is *ideal-like* as defined by Curto and Fialkow [14]. We note that there is a bijective correspondence between vectors  $v \in \ker M(\lambda_d)$  and polynomials given by  $v \mapsto \mathbb{P}(v) = v^T (x^\alpha)_{\alpha \in \mathbb{N}^n}$  where  $(x^\alpha)_{\alpha \in \mathbb{N}^n}$  is the vector of all monomials of degree  $\leq d$  ordered in the same way as the rows of the moment matrix. Conversely each polynomial  $g$  used to form the coefficient matrix  $B$ , is mapped to a vector  $\text{vec}(g)$  in  $\ker M(\lambda_d)$ .

**Definition 4.7.1 (Ideal-Like truncated moment matrix [14])** *The kernel of a truncated moment matrix  $M(\lambda_d)$  is ideal-like of degree  $d$  if the following two conditions are satisfied:*

- *If  $f_1, f_2 \in \mathbb{P} \ker M(\lambda_d)$  then  $f_1 + f_2 \in \mathbb{P} \ker M(\lambda_d)$ .*
- *If  $f \in \mathbb{P} \ker M(\lambda_d)$  and  $g \in \mathbb{R}[x]$  has  $\deg(fg) \leq d$ , then  $fg \in \mathbb{P} \ker M(\lambda_d)$ .*

*The ideal-like property is denoted as RG in [14].*

Our main result is:

**Theorem 4.7.1** *Suppose that  $I = \langle f_1, \dots, f_m \rangle_{\mathbb{R}}$  with  $\max_i(\deg(f_i)) = d$  and let  $B$  be the coefficient matrix of  $\{f_1, \dots, f_m\} \subseteq \mathbb{R}[x]$ . Let  $M(\lambda_d)$  be a truncated moment matrix such that  $B \cdot M(\lambda_d) = 0$  and  $M(\lambda_d) \geq 0$ . If the rank of  $M(\lambda_d)$  is maximum and  $\ker M(\lambda_d)$  is ideal-like then*

$$\mathbb{P} \ker M(\lambda_d) = \left(\sqrt[\mathbb{R}]{I}\right)_{(\leq d)} \quad (4.25)$$

To prove the above theorem, we will need Theorem 4.7.2, Theorem 4.7.3 and Lemma 4.7.4 below.

**Theorem 4.7.2** [27, Lemma 3.1] Suppose that the ideal  $I = \langle f_1, \dots, f_m \rangle_{\mathbb{R}}$  with  $\max_i(\deg(f_i)) = d$  and let  $B$  be the coefficient matrix of  $\{f_1, \dots, f_m\} \subseteq \mathbb{R}[x]$ . Let  $M(\lambda_d)$  be a truncated moment matrix such that  $B \cdot M(\lambda_d) = 0$  and  $M(\lambda_d) \geq 0$ . If the rank of  $M(\lambda_d)$  is maximum then

$$\mathbb{P} \ker M(\lambda_d) \subseteq \sqrt[d]{I} \quad (4.26)$$

**Theorem 4.7.3** (Flat extension theorem [14]) Assume  $M(\lambda_d) \geq 0$ . The following statements are equivalent:

(i) There exists an extension  $M(\lambda_{d+1}) \geq 0$  and  $\text{rank } M(\lambda_d) = \text{rank } M(\lambda_{d+1})$

(ii)  $\ker M(\lambda_d)$  is ideal-like.

**Lemma 4.7.4** [27, Theorem 3.4, Corollary 3.8] Assume  $M(\lambda) \geq 0$  and  $\text{rank } M(\lambda_d) = \text{rank } M(\lambda_{d-1}) = r$ . Then  $J = \langle \mathbb{P} \ker M(\lambda_d) \rangle_{\mathbb{R}}$  is real radical and zero-dimensional. One can extend  $\lambda_d$  to  $\lambda = \sum_{i=1}^r \alpha_i \lambda_{v_i} \in \mathbb{R}[x]^*$  where  $\alpha_i > 0$  and  $\{v_1, \dots, v_r\} = V_{\mathbb{R}}(\mathbb{P} \ker M(\lambda_d))$ . Furthermore  $\lambda = \lambda_d$  when  $\lambda$  is restricted to  $\mathbb{R}[x]_{2d}$ .

We now prove Theorem 4.7.1.

**Proof** Suppose  $\ker M(\lambda_d)$  is ideal-like,  $M(\lambda_d) \geq 0$  and  $M(\lambda_d)$  has maximum rank together with the other assumptions in Theorem 4.7.1.

Our goal is to show that

$$\mathbb{P} \ker M(\lambda_d) = \left( \sqrt[d]{I} \right)_{(\leq d)}.$$

First by Theorem 4.7.2, the following direction is obvious:

$$\mathbb{P} \ker M(\lambda_d) \subseteq \left( \sqrt[d]{I} \right)_{(\leq d)}.$$

So we only need to show

$$\mathbb{P} \ker M(\lambda_d) \supseteq \left( \sqrt[d]{I} \right)_{(\leq d)}$$

By Theorems 4.7.3 and 4.7.4,  $\lambda_d$  can be extended to  $\lambda_{d+1}$  such that  $J = \langle \mathbb{P} \ker M(\lambda_{d+1}) \rangle_{\mathbb{R}}$  is real radical and zero-dimensional. Since  $I \subseteq J$ , we have  $\sqrt[d]{I} \subseteq J$ . By Theorem 4.7.4, one can extend  $\lambda_d$  to  $\lambda = \sum_{i=1}^r \alpha_i \lambda_{v_i} \in \mathbb{R}[x]^*$  where  $\alpha_i > 0$  and  $\{v_1, \dots, v_r\} = V_{\mathbb{R}}(\mathbb{P} \ker M(\lambda_{d+1})) = V_{\mathbb{R}}(J)$

and  $\lambda_{v_i}$  is an evaluation mapping at  $v_i$  such that  $\lambda_{v_i}(f) = f(v_i)$ . Thus  $\lambda_d = \sum_{i=1}^r \alpha_i \lambda_{v_i}^{(d)}$  where  $\lambda_{v_i}^{(d)}$  is the truncated linear form of  $\lambda_{v_i}$ . Since  $\sqrt[r]{I} \subseteq J$ , we have  $\{v_1, \dots, v_r\} \subseteq V_{\mathbb{R}}(\sqrt[r]{I})$ .

Now we can prove the other inclusion:

$$\mathbb{P} \ker M(\lambda_d) \supseteq \left( \sqrt[r]{I} \right)_{(\leq d)}$$

So we let  $g \in \left( \sqrt[r]{I} \right)_{(\leq d)}$  and we want to show that  $g \in \mathbb{P} \ker M(\lambda_d)$ , that is to show that  $\text{vec}(g)^T M(\lambda_d) = 0$ .

Since  $g \in \sqrt[r]{I}$  with  $\deg(g) \leq d$ , we have  $g(v_i) = 0, i = 1, \dots, r$ . Therefore  $g^2(v_i) = \text{vec}(g)^T M(\lambda_{v_i}^{(d)}) \text{vec}(g) = 0$ . Since  $M(\lambda_{v_i}^{(d)}) \geq 0$ , we have  $\text{vec}(g)^T M(\lambda_{v_i}) = 0$  for  $i = 1, \dots, r$ . Hence  $\sum_{i=1}^r \alpha_i \text{vec}(g)^T M(\lambda_{v_i}^{(d)}) = 0$ , so  $\text{vec}(g)^T M(\lambda_d) = 0$  and  $g \in \mathbb{P} \ker M(\lambda_d)$  which is what we wanted to show. ■

By Theorem 4.7.1, we now have a complete algorithm to Problem 4.7.1

---

**Algorithm 4.7.1:** RealRadical( $F, d$ )

---

- 1 Input( $F = \{f_1, \dots, f_m\} \subseteq \mathbb{R}[x]$ ,  $x \in \mathbb{R}^n$ , an integer  $d \geq \deg(F)$ .);
  - 2 Set  $F'$  to the prolongation of  $F$  to degree  $d$
  - 3 **repeat**
  - 4      $B := \text{CoeffMtx}(F')$
  - 5     Solve Problem 4.2.1 for maximum rank moment matrix  $M(\lambda_d)$  by Algorithm 4.3.1.
  - 6      $F'' := \mathbb{P}(\ker M(\Lambda_d))$
  - 7     Compute  $\text{GIF}(F'')$
  - 8     Project/ Prolong  $\text{GIF}(F'')$  to degree  $d$ :  $F' := \text{GIF}(F'')_{(\leq d)}$ .
  - 9 **until**  $\dim F' = \dim F''$ ;
  - 10 Output( $F'$ , a basis for  $\{f \in \sqrt[r]{I} : \deg(f) \leq d\}$ )
- 

In Algorithm 4.7.1,  $\text{CoeffMtx}$  computes the coefficients in the monomial basis, although potentially other bases could be used. It exploits the property that the the GIF algorithm obtains polynomials in a form that satisfies the ideal-like property. In particular note that for a given  $f$  in Definition 4.7.1,  $fg = \sum_{\alpha} a_{\alpha} x^{\alpha} f$  is expanded in term of so-called prolongations by monomials  $x^{\alpha}$ . The invariance of geometric involutive bases under prolongation-projection

implies that each  $x^\alpha f$  is in the basis, and by superposition  $fg$  is also in the basis. We note that Pommaret involutive bases don't necessarily satisfy the ideal-like property but can be extended easily by an explicit algorithm to such basis [21, 39]. Groebner bases can also be extended, by essentially reformulating them as involutive basis [21].

Involutivity originates in the geometry of differential equations. See Kuranishi [25] for a famous proof of termination of Cartan's prolongation algorithm for nonlinear partial differential equations. A by-product of these methods has been their implementation for linear homogeneous partial differential equations with constant coefficients, and consequently for polynomial algebraic systems. See [21] for applications and symbolic algorithms for polynomial systems. The symbolic-numeric version of a *geometric involutive form*, *GIF*, was first described and implemented in Wittkopf and Reid [46]. It was applied to approximate symmetries of differential equations in [6] and to polynomial solving in [37, 35, 38]. See [49] where it is applied to the deflation of multiplicities in multivariate polynomial solving. For more details and examples see [36, 6]. The details of the GIF algorithm, including, prolongations and projections, can be found in our earlier work [34] and in chapter 2.

An easy consequence is that the result also applies to the output of our GIF-FDR algorithm.

**Theorem 4.7.5** *Let  $F = \{f_1, \dots, f_m\} \subset \mathbb{R}[x]$ . Let  $G = \{g_1, \dots, g_k\} \subset \mathbb{R}[x]$  be the output of the GIF-FDR algorithm applied to  $F$ . Then*

$$\left( \sqrt[\mathbb{R}]{\langle F \rangle_{\mathbb{R}}} \right)_{(\leq d)} = \text{span}_{\mathbb{R}} G, \quad d = \deg(G) \quad (4.27)$$

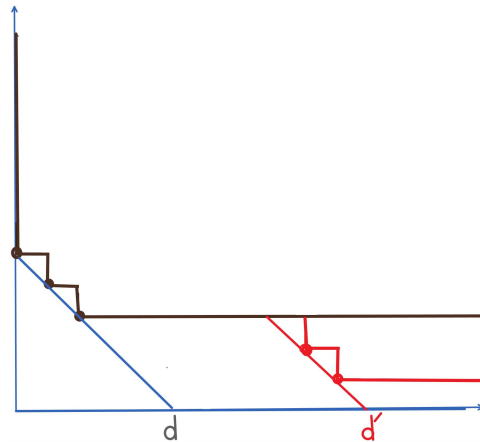
In the 0-dimensional case, we also have the following theorem:

**Theorem 4.7.6** *Let  $F = \{f_1, \dots, f_m\} \subset \mathbb{R}[x]$ . Let  $G = \{g_1, \dots, g_k\} \subset \mathbb{R}[x]$  be the output of the GIF-FDR algorithm applied to  $F$  and the Hilbert dimension of  $\langle G \rangle_{\mathbb{R}}$  is 0, i.e., the system  $G$  has finitely many complex solutions. Then*

$$\sqrt[\mathbb{R}]{\langle F \rangle_{\mathbb{R}}} = \langle G \rangle_{\mathbb{R}} \quad (4.28)$$

**Proof** From the Algorithm GIF-FDR, we know that  $G$  is already involutive. Also because the Hilbert dimension of  $\langle G \rangle_{\mathbb{R}}$  is zero, any monomial of degree not less than  $d = \deg(G)$  is one

## 4.8 A special case for determining positive dimensional real radical



Our theorem on the determination of the real radical up to finite degree is illustrated graphically in Figure 4.1. Here suppose  $F = \{f_1, \dots, f_m\} \subset \mathbb{R}[x]$  and we applied Algorithm  $\text{RealRadical}(F, d)$  for a given  $d$ , and that the resulting system has leading monomials shown as the corners of the black monomial staircase. See [13] for the description of such diagrams. Then the system is prolonged and the kernel of its moment matrix is examined for new generators at degrees  $d + 1, d + 2, \dots$ . The only way that this is not a complete generating set for

the real radical (and that our conjecture fails), is that there is a minimum degree  $d' > d$  where after prolongation to  $d'$  new generators are determined that lie outside simple prolongations of the black leading generators. These have leading monomials shown in red. Some times the completeness of the generating set at degree  $d$  can be checked by a critical point calculation. For example, if the critical point method shows that the variety is real positive dimensional, then this could rule out the existence of the red staircase predicting a 0-dimensional real variety. In particular, if the number of red circles in Figure 4.1 is 1 and the variety of  $F$  is real positive dimensional, then  $\text{RealRadical}(F, d)$  returns the generators of  $\sqrt[d]{\langle F \rangle_{\mathbb{R}}}$ . So we have the following theorem:

**Theorem 4.8.1** *Given a system of polynomials  $F = \{f_1, \dots, f_m\} \subseteq \mathbb{R}[x_1, x_2, \dots, x_n]$  with associated ideal  $I$  and an integer  $d$ . Let  $G = \{g_1, \dots, g_k\} \subset \mathbb{R}[x]$  be the output of the  $\text{RealRadical}(F, d)$  algorithm applied to  $F$  and  $s$  is the number of different polynomials of degree  $d$  in  $G$ . If  $s = \binom{d+n-1}{n-1} - 1$  and the variety of  $F$  is real positive dimensional. Then*

$$\sqrt[d]{\langle F \rangle_{\mathbb{R}}} = \langle G \rangle_{\mathbb{R}}. \quad (4.29)$$

**Proof** By Theorem 4.7.1,  $(\sqrt[d]{\langle F \rangle_{\mathbb{R}}})_{(\leq d)} = \text{span}_{\mathbb{R}} G$ . Suppose in contradiction  $\sqrt[d]{\langle F \rangle_{\mathbb{R}}} \supset \langle G \rangle_{\mathbb{R}}$ , then there exists a  $d' > d$  such that  $(\langle H \rangle_{\mathbb{R}})_{(\leq d')} \subset (\sqrt[d]{\langle F \rangle_{\mathbb{R}}})_{(\leq d')}$  where  $H$  is the prolongation of  $G$  to degree  $d'$ . Therefore there exists a polynomial  $\tilde{g} \in \text{span}_{\mathbb{R}} \tilde{G}$  but  $g \notin \text{span}_{\mathbb{R}} H$  with  $\deg(\tilde{g}) = d' > d$  where  $\tilde{G} = \{\tilde{g}_1, \dots, \tilde{g}_l\}$  spans  $(\sqrt[d]{\langle F \rangle_{\mathbb{R}}})_{(\leq d')}$ .

Now assume the number of different polynomials of degree  $d'$  in  $H$  is  $t$  and the number of different polynomials of degree  $d'$  in  $\tilde{G}$  is  $\bar{t}$ , then  $t < \bar{t}$  because the existence of  $\tilde{g}$ . From combinatorics, the number of different monomials of degree  $d$  in  $n$  variables is  $\binom{d+n-1}{n-1}$ . Since  $G$  is already involutive and  $s = \binom{d+n-1}{n-1} - 1$ , we have  $t = \binom{d'+n-1}{n-1} - 1$  as well. Also clearly  $\bar{t} \leq \binom{d'+n-1}{n-1}$ , so we have  $\bar{t} = \binom{d'+n-1}{n-1}$  which means  $\sqrt[d]{\langle F \rangle_{\mathbb{R}}}$  is a 0-dimensional real variety, a contradiction with the assumption that the variety of  $F$  is real positive dimensional. So the theorem is proved. ■

## 4.9 Comparison with Triangular decomposition of semi-algebraic sets

In this section, we compare our method with the triangular decomposition of semi-algebraic sets.

One of the motivations for computing the real radical ideal is to remove the multiplicities and sum of squares of a given polynomial system. Our method in this thesis is a “global” method, i.e., we don’t compute each connected component of the real variety. The triangular decomposition of semi-algebraic sets is a local method, i.e., it computes an intersection of primal ideals in the real fields while each primal ideal represents a connected component of the real variety.

**Example 4.9.1** [10]

$$f = \{2yz - y, 2y^2 + y, xy, 4x^2z + 4z^3 + y\} \quad (4.30)$$

By using real triangular decomposition, we obtained an intersection of three primary ideals:  $\{x, y, z\} \cap \{y, z\} \cap \{x, 2y + 1, 2z - 1\}$ . By our approach, we obtained the generators  $\{z^2 + y/2, yz - y/2, y^2 + y/2, xz, xy, y + z\}$

Also, our method is stable under small perturbations. If we add a small perturbation to the above polynomial system,

**Example 4.9.2**

$$f = \{2yz - y, 2y^2 + y + 10^{-13}, xy, 4x^2z + 4z^3 + y\} \quad (4.31)$$

By using real triangular decomposition, we obtained only one primary ideals:  $\{x, y + 0.5 + \epsilon, 2z - 1\}$ . By our approach, we obtained perturbed generators  $\{z^2 + y/2 + \epsilon, yz - y/2 + \epsilon, y^2 + y/2 + \epsilon, xz + \epsilon, xy + \epsilon, y + z + \epsilon\}$

## 4.10 Examples

In this section, we give some examples. We used MATLAB version 2015a. The computations were carried out on a desktop with ubuntu 12.04 LTS, Intel Core™2 Quad CPU Q9550 @ 2.83 GHz  $\times$  4, 8GB RAM, 64-bit OS, x64-based processor.

We give the first examples (Ex.4.33 and Ex.4.34) showing additional facial reductions for polynomials, that can be accurately approximated in practice.

**Example 4.10.1 (Reducible cubic)**

$$(x + y)(x^2 + y^2 + 2) \quad (4.32)$$

*Note that the second factor has no real roots, so it is discarded and the real radical is generated by  $(x+y)$ . The moment matrix corresponding to (4.32) is a  $10 \times 10$  matrix. The coefficient matrix  $B$  is  $[0, 2, 2, 0, 0, 0, 1, 1, 1, 1]^T$ . Using Algorithm 4.7.1, after two facial reductions, we obtained a maximum rank 4 moment matrix with residual less than  $10^{-14}$  in less than 200 DR iterations and the generators of real radical is computed to degree 3. The GIF-FDR algorithm correctly yields to high accuracy the generator  $(x + y)$  of the real radical to degree 1 as predicted by Theorem 4.7.5.*

*We compare it with SeDuMi(CVX), SeDuMi(CVX) obtains a rank 4 moment matrix with 9 decimal accuracy without maximizing the rank. However if we maximize the rank (by maximizing the trace which is used in other examples as well) in CVX, the accuracy is only to 2 decimal places.*

**Example 4.10.2 (Reducible quintic)**

$$(1 + x + y)(x^4 + y^4 + 2) \quad (4.33)$$

*The moment matrix corresponding to (4.33) is a  $21 \times 21$  matrix. We solve this problem using Algorithm 4.7.1. Algorithm 4.7.1 can get 14 decimal accuracy and a maximum rank moment matrix of rank 6 in about 1300 DR iterations with 2 facial reductions. The output approximates the real radical ideal generated by  $\langle 1 + x + y \rangle$  and its prolongations to degree 5. The GIF-FDR algorithm obtains the correct real radical generator  $(1 + x + y)$  to degree 1 as predicted by Theorem 4.7.5.*

*We compare it with SeDuMi(CVX). SeDuMi(CVX) can get a rank 6 moment matrix with 13 decimal accuracy without maximizing the rank. However if we maximize the rank in CVX, we only get 9 decimal accuracy.*



**Example 4.10.3 (Two variable geometric polynomial with 3 facial reductions)**

$$1 + (x + y) + (x + y)^2 + (x + y)^3 \quad (4.34)$$

The moment matrix corresponding to (4.34) is a  $10 \times 10$  matrix. The coefficient matrix  $B$  is  $[2, 2, 2, 1, 0, 1, 1, 1, 1, 1]^T$ .

This example is a demonstration of the ill-conditioned case discussed in Section 4.5. We first solve it using Algorithm 4.3.1 with rank  $r$  to be maximum in  $\mathcal{P}_{S_+^k}(X, r)$ , which returns solution of rank 5 with residual  $10^{-7}$  after 2 facial reductions. However, the DR method for solving the auxiliary problem (4.9b) converges very slowly. So we check the eigenvalues of solution of the auxiliary problem (4.9b). After the first facial reduction, the eigenvalues are 0.5, 0.2, 0.18, 0.08, 0, 0, 0, 0, 0. So we drop the fourth one and set  $r = 3$ . We resolve (4.9b) using the DR method, which again is quite slow. So we check the eigenvalues and they are now 0.709, 0.29, 0.00002, 0, 0, 0, 0, 0, 0. The third one is very small so we drop it and set  $r = 2$ . Then we resolve (4.9b) with  $r = 2$ . This time the auxiliary problem is solved with residual  $10^{-15}$ . Then a third facial reduction is done by setting  $r = 3$  and the residual is  $10^{-14}$ .

After 3 facial reductions, the face is reduced to dimension 4 and the moment matrix is obtained with residual  $10^{-13}$ . The eigenvalues of the final moment matrix are 4.70, 3.48, 0.89, 0.59, 0, 0, 0, 0, 0 which gives the correct maximum rank of 4.

We compare it with SeDuMi(CVX) SDP solver. If we maximize the rank in CVX, we can obtain a moment matrix with residual about  $10^{-9}$ , the moment matrix has 8 positive eigenvalues and the 5th eigenvalue is  $3 \times 10^{-5}$ . So in order to get the correct maximum rank, the threshold has to be set to  $10^{-4}$  which is not accurate. If we do not maximize the rank, the residual is similar only the threshold is slightly better which is  $10^{-5}$ .

This example involves 3 facial reductions, the size of the problem after each facial reduction is 10, 9, 7, 4. Actually, this example has singularity degree 2 if we don't count the first "trivial" facial reduction. If we set the rank to be 5 when solving the auxiliary problem, it only returns a solution of rank 4 meaning we can't reduce the problem to the minimal face by solving the auxiliary problem only once. We tried the DR method to maximize the rank of the auxiliary problem with random initial values 100 times, all yielding solutions of rank 4.

Actually we can prove the singularity is more than 1. We know the real radical of this

polynomial system is  $\{1 + x + y, x + x^2 + xy, y + xy + y^2, x^2 + x^3 + x^2y, xy + x^2y + xy^2, y^2 + xy^2 + y^3\}$  to degree 3. Let  $N$  be the coefficient matrix of this polynomial system. Then  $Q = V^T N N^T V$  will be the orthogonal complement of the primal problem  $\bar{\mathcal{A}}(X) = \bar{b}, X \geq 0$  with rank 5 where  $V^T B = 0$ . If the singularity degree is 1, then  $Q = \sum_{i=1}^m \bar{A}_i y_i$  must be consistent ( $\bar{b}^T y = 0 \implies y_0 = 0$ ). By checking the rank of  $[\bar{\mathcal{A}}, \text{s2vec}(Q)]$  and  $\bar{\mathcal{A}}$ , we found the linear system is inconsistent so the singularity degree is 2.

Application of Algorithm 4.7.1 yields the correct generators of the real radical up to degree 3. Application of GIF-FDR algorithm yields the generators of real radical to degree 1 which is  $1 + x + y$ .

#### Example 4.10.4 [10]

$$f = \{2yz - y, 2y^2 + y, xy, 4x^2z + 4z^3 + y\} \quad (4.35)$$

The real radical of this polynomial system is [10]:

$$\{z^2 + y/2, yz - y/2, y^2 + y/2, xz, xy, y + z\}$$

The moment matrix of this problem is  $20 \times 20$ . We use Algorithm 4.3.1 to solve for maximum rank moment matrix. The sizes of the SDP problem are  $[20, 16, 14, 8]$  after 3 facial reductions. The residual of the auxiliary problem at each facial reduction is  $10^{-15}, 10^{-14}$ . (The first facial reduction is done by Matlab eigenvalue decomposition so we don't put its residual here.) The moment matrix is solved with residual  $10^{-13}$  and the maximum rank is 8.

We compare it with SeDuMi(CVX) which shows very poor performance. If we maximize the rank in CVX, the residual of the moment matrix solved by SeDuMi(CVX) is  $8.5 \times 10^{-11}$  with 9 positive eigenvalues, of which 6 eigenvalues are greater than 0.1 and the other three eigenvalues are around  $5 \times 10^{-7}$ . If we do not maximize the rank in CVX, then the residual is  $8 \times 10^{-10}$ . But to get the correct rank, the threshold for the eigenvalues has to be set to  $1 \times 10^{-7}$ . So in general, it is very difficult to use SeDuMi(CVX) to get the correct maximum rank.

As the computations in the above examples and Table 4.1, 4.2 demonstrate, the traditional interior point SDP solver SeDuMi(CVX) is not the right choice for computing the maximum rank moment matrices as it usually yields poorer performance when it is trying to maximize

	min # FR	max # FR	rank (FR)	Singlty deg	Res(FR)	Res(CVX)
Ex 4.32	2	3	10, 9, 4	1	$10^{-14}$	$10^{-9}$
Ex 4.33	2	unknown	21, 20, 6	1	$10^{-14}$	$10^{-9}$
Ex 4.34	3	4	10, 9, 7, 4	2	$10^{-13}$	$10^{-9}$
Ex 4.10.4	3	4	20, 16, 14, 8	2	$10^{-13}$	$10^{-9}$

**Table 4.1: Comparison between facial reduction and SeDuMi (1)** All data is obtained by using minimal number of facial reductions; Here: min (max) # FR means minimal (maximum) number of facial reductions in our tests; rank(FR) means the size of the problem after each facial reduction, the first one is the size of the original problem; Singlty degree is the singularity degree of the SDP problem after the 1st facial reduction; Res(FR) is the residual of the final moment matrix using facial reduction and DR iterations (Algorithm 4.3.1); Res(CVX) is the residual of the final moment matrix using CVX(SeDuMi).

	max rank	res each FR	# DR each FR	thres FR	thres CVX
Ex 4.32	4	$10^{-15}, 10^{-15}$	120, 7	$10^{-16}$	$10^{-12}$
Ex 4.33	6	$10^{-15}, 10^{-14}$	267, 6	$10^{-16}$	$10^{-9}$
Ex 4.34	4	$10^{-15}, 10^{-14}, 10^{-15}$	260, 143, 1	$10^{-16}$	$10^{-5}$
Ex 4.10.4	8	$10^{-15}, 10^{-14}, 10^{-14}$	625, 192, 29	$10^{-16}$	$10^{-7}$

**Table 4.2: Comparison between facial reduction and SeDuMi (2)** All data obtained here is by using minimal number of facial reductions; max rank is the maximum rank of the moment matrix; res each FR is the residual of solving the corresponding SDP problem by DR after each facial reduction; # DR each FR is the number of DR iterations to solve the corresponding SDP problem after each facial reduction; thres FR is the tolerance to obtain the correct maximum rank using facial reductions (Algorithm 4.3.1); thres CVX is the tolerance to obtain the correct maximum rank using CVX(SeDuMi);

rank. It even gets better performance without maximizing the rank! With facial reductions and the DR method, we can get much better accuracy and also the correct maximum rank.

In the above examples, Algorithm 4.7.1 and GIF-FDR follow the same path except that GIF-FDR executes an extra step which reduces the degree of the output. Generally, however, the paths of these two algorithms can be quite different.

## 4.11 Conclusion

SDP feasibility problems typically involve the intersection of the convex cone of semi-definite matrices with a linear manifold. Their importance in applications has led to the development of many specific algorithms. However these feasibility problems are often marginally infeasible, i.e., they do not satisfy strict feasibility as is the case for our polynomial applications. Such problems are *ill-posed* and *ill-conditioned*.

This chapter is part of a series in which we exploit facial reduction and its application systems of real polynomial and differential equations for real solutions. The current work is directed at guaranteeing the maximal rank property and the ideal-like condition to ensure all the generators of the real radical up to a given degree are captured. It also establishes the first examples of additional facial reduction that are effective in practice for polynomial systems.

This builds on our work in [34] in which we introduced facial reduction, for the class of SDP problems arising from analysis and solution of systems of real polynomial equations for real solutions. Facial reduction yields an equivalent smaller problem for which there are strictly feasible generic points. Facial reduction also reduces the size of the moment matrices occurring in the application of SDP methods. For example the determination of a  $k \times k$  moment matrix for a problem with  $m$  linearly independent constraints is reduced to a  $(k - m) \times (k - m)$  moment matrix by one facial reduction. The high accuracy required by facial reduction and also the ill-conditioning commonly encountered in numerical polynomial algebra [43] motivated us to implement Douglas-Rachford iteration in [34].

A fundamental open problem is to generalize the work of [26, 42] to positive dimensional ideals. The algorithm of [32, 31] for a given input real polynomial system  $P$ , modulo the

successful application of SDP methods at each of its steps, computes a Pommaret basis  $Q$ :

$$\sqrt[\mathbb{R}]{\langle P \rangle_{\mathbb{R}}} \supseteq \langle Q \rangle_{\mathbb{R}} \supseteq \langle P \rangle_{\mathbb{R}} \quad (4.36)$$

and would provide a solution to this open problem if it is proved that  $\langle Q \rangle_{\mathbb{R}} = \sqrt[\mathbb{R}]{\langle P \rangle_{\mathbb{R}}}$ . We believe that the work [32, 31] establishes an important feature – involutivity – that will necessarily be a main condition of any theorem and algorithm characterizing the real radical. Involutivity is a natural condition, since any solution of the above open problem using SDP, if it establishes radical ideal membership, will necessarily need (at least implicitly) a real radical Gröbner basis. Our algorithm, uses geometric involutivity, and similarly gives an intermediate ideal, which constitutes another variation on this family of conjectures.

An important open problem is the following: *Give an numerical algorithm, capable in principle of determining an approximate real point on each component of a real variety.* We note that the methods of Wu and Reid [48] and Hauenstein [22] only answer this question under certain conditions, say that the ideal is real radical and defined by a regular sequence. Also see [30], which gives an alternative extension of complex numerical algebraic geometry to the reals, in the complex curve case.

Recently, Hauenstein et al [10] have made progress on this problem by using sample points determined by Hauenstein’s critical point algorithm which is able to certify the generators of the real radical ideal in some cases. Our results Theorem 4.7.1 and Theorem 4.7.5 enables the determination of the generators up to a given degree. Thus gives an answer to the open problem of real radical ideal membership test left in [10]. Potentially, the efficiency for computing the sample points can also be improved which will be described in a subsequent work.

## Bibliography

- [1] A. Alfakih and H. Wolkowicz. Matrix completion problems. In *Handbook of semidefinite programming*, volume 27 of *Internat. Ser. Oper. Res. Management Sci.*, pages 533–545. Kluwer Acad. Publ., Boston, MA, 2000. 54, 94
- [2] F.J.A. Artacho, J.M. Borwein, and M.K. Tam. Recent results on Douglas-Rachford methods. *Serdica Mathematical Journal*, 39:313–330, 2013. 73, 105

- [3] H.H. Bauschke, J.Y. Bello Cruz, T.T.A. Nghia, H.M. Phan, and X. Wang. The rate of linear convergence of the Douglas-Rachford algorithm for subspaces is the cosine of the Friedrichs angle. *J. Approx. Theory*, 185:63–79, 2014. 75, 105
- [4] H.H. Bauschke and D. Noll. On the local convergence of the Douglas-Rachford algorithm. *Arch. Math. (Basel)*, 102(6):589–600, 2014. 75, 105
- [5] G. Blekherman, P.A. Parrilo, and R.R. Thomas, editors. *Semidefinite optimization and convex algebraic geometry*, volume 13 of *MOS-SIAM Series on Optimization*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA, 2013. 25, 55, 56, 68, 94
- [6] J. Bonasia, F. Lemaire, G.J. Reid, and L. Zhi. Determination of approximate symmetries of differential equations. *Group Theory and Numerical Analysis*, 39:249, 2005. 24, 31, 33, 35, 59, 61, 112
- [7] J.M. Borwein and M.K. Tam. A Cyclic Douglas–Rachford Iteration Scheme. *J. Optim. Theory Appl.*, 160(1):1–29, 2014. 73, 74, 105
- [8] J.M. Borwein and H. Wolkowicz. Facial reduction for a cone-convex programming problem. *J. Austral. Math. Soc. Ser. A*, 30(3):369–380, 1980/81. 12, 54, 94, 98, 106
- [9] J.M. Borwein and H. Wolkowicz. Regularizing the abstract convex program. *J. Math. Anal. Appl.*, 83(2):495–530, 1981. 12, 54, 94, 98, 106
- [10] D. Brake, J. Hauenstein, and A. Liddell. Numerically validating the completeness of the real solution set of a system of polynomial equations. *Proceedings of the 41th International Symposium on Symbolic and Algebraic Computation*, 2016. 1, 115, 118, 121, 130
- [11] Y-L. Cheung, S. Schurr, and H. Wolkowicz. Preprocessing and regularization for degenerate semidefinite programs. In D.H. Bailey, H.H. Bauschke, P. Borwein, F. Garvan, M. Thera, J. Vanderwerff, and H. Wolkowicz, editors, *Computational and Analytical Mathematics, In Honor of Jonathan Borwein’s 60th Birthday*, volume 50 of *Springer Proceedings in Mathematics & Statistics*, pages 225–276. Springer, 2013. 12, 70, 72, 97, 98

- [12] Y.-L. Cheung and H. Wolkowicz. Sensitivity analysis of semidefinite programs without strong duality. Technical report, University of Waterloo, Waterloo, Ontario, 2014. submitted June 2014, 37 pages. 54, 94
- [13] David Cox, John Little, and Donal O’shea. *Ideals, varieties, and algorithms*, volume 3. Springer, 1992. 2, 7, 113
- [14] RE Curto and LA Fialkow. Solution of the truncated complex moment problem for flat data-introduction. *Memoirs of the American Mathematical Society*, 119(568):1, 1996. 13, 15, 16, 17, 25, 108, 109, 110
- [15] Jim Douglas, JR and H.H. Rachford, JR. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Amer. Math. Soc.*, 82:421–439, 1956. 73, 105
- [16] D. Drusvyatskiy, N. Krislock, Y-L. Cheung Voronin, and H. Wolkowicz. Noisy sensor network localization: robust facial reduction and the Pareto frontier. Technical report, University of Waterloo, Waterloo, Ontario, 2014. arXiv:1410.6852, 20 pages. 12, 54, 94
- [17] D. Drusvyatskiy, G. Li, and H. Wolkowicz. Alternating projections for ill-posed semidefinite feasibility problems. Technical report, University of Waterloo, Waterloo, Ontario, 2014. submitted Sept. 2014, 12 pages. 54, 69, 94, 102
- [18] D. Drusvyatskiy, G. Pataki, and H. Wolkowicz. Coordinate shadows of semidefinite and euclidean distance matrices. *Math. Programming*, 25(2):1160–1178, 2015. ArXiv:1405.2037.v1. 12, 98
- [19] M. Dür, B. Jargalsaikhan, and G. Still. The Slater condition is generic in linear conic programming. Technical report, University of Trier, Trier, Germany, 2012. 54, 94
- [20] R. Escalante and M. Raydan. *Alternating projection methods*, volume 8 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2011. 72, 103

- [21] V.P. Gerdt and Y.A. Blinkov. Involutive bases of polynomial ideals. *Mathematics and Computers in Simulation*, 45(5):519–541, 1998. 7, 24, 29, 59, 112
- [22] Jonathan D Hauenstein. Numerically computing real points on algebraic sets. *Acta applicandae mathematicae*, 125(1):105–119, 2013. 5, 26, 83, 85, 121
- [23] Nicholas J. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications*, 103:103 – 118, 1988. 103
- [24] N. Krislock and H. Wolkowicz. Explicit sensor network localization using semidefinite representations and facial reductions. *SIAM Journal on Optimization*, 20(5):2679–2708, 2010. 12, 54, 94
- [25] M. Kuranishi. On e. cartan’s prolongation theorem of exterior differential systems. *American Journal of Mathematics*, pages 1–47, 1957. 24, 29, 33, 59, 112
- [26] J.B. Lasserre, M. Laurent, and P. Rostalski. A prolongation–projection algorithm for computing the finite real variety of an ideal. *Theoretical Computer Science*, 410(27):2685–2700, 2009. 1, 8, 25, 47, 53, 54, 55, 64, 84, 93, 94, 120
- [27] Jean Bernard Lasserre, Monique Laurent, and Philipp Rostalski. Semidefinite characterization and computation of zero-dimensional real radical ideals. *Foundations of Computational Mathematics*, 8(5):607–647, 2008. 13, 108, 110, 128
- [28] M. Laurent and P. Rostalski. The approach of moments for polynomial equations. In Miguel F. Anjos and Jean B. Lasserre, editors, *Handbook on semidefinite, conic and polynomial optimization*, International Series in Operations Research & Management Science, 166, pages 25–60. Springer, New York, 2012. 10, 13, 14, 15, 16, 17, 36, 38, 40, 46, 47, 58, 95, 96, 128, 129
- [29] P. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979. 73, 105
- [30] Y. Lu, D.J. Bates, A.J. Sommese, and C.W. Wampler. Finding all real points of a complex curve. In *Algebra, geometry and their interactions*, volume 448 of *Contemp. Math.*, pages 183–205. Amer. Math. Soc., Providence, RI, 2007. 85, 121



- [31] Y. Ma. *Polynomial Optimization via Low-rank Matrix Completion and Semidefinite Programming*. PhD thesis, Academy of Mathematics and Systems Science, Chinese Academy of Science, 2012. 8, 53, 55, 62, 77, 84, 93, 120, 121
- [32] Y. Ma, C. Wang, and L. Zhi. A certificate for semidefinite relaxations in computing positive dimensional real varieties. *Journal of Symbolic Computation*, 72:1 – 20, 2016. vii, x, 1, 8, 53, 55, 62, 77, 78, 79, 80, 81, 82, 84, 93, 120, 121, 128, 129
- [33] Carl D Meyer. *Matrix analysis and applied linear algebra*, volume 2. Siam, 2000. 104
- [34] G. Reid, F. Wang, H. Wolkowicz, and W. Wu. Semidefinite Programming and facial reduction for Systems of Polynomial Equations. *Preprint arXiv:1504.00931v1*, 2015. 1, 93, 95, 96, 108, 112, 120
- [35] G.J. Reid, J. Tang, and L. Zhi. A complete symbolic-numeric linear method for camera pose determination. In *Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, pages 215–223. ACM, 2003. 24, 31, 33, 59, 112
- [36] G.J. Reid, F. Wang, and W. Wu. Geometric involutive bases for positive dimensional polynomial ideals and sdp methods. Technical report, Department of Appl. Math., University of Western Ontario, 2014. 57, 59, 60, 62, 112
- [37] G.J. Reid and L. Zhi. Solving polynomial systems via symbolic-numeric reduction to geometric involutive form. *Journal of Symbolic Computation*, 44(3):280–291, 2009. 24, 59, 112
- [38] R. Scott, G.J. Reid, W. Wu, and L. Zhi. Geometric involutive bases and applications to approximate commutative algebra. In Lorenzo Robbiano and John Abbott, editors, *Approximate Commutative Algebra*, pages 99–124. Springer, 2010. 24, 37, 43, 55, 59, 60, 95, 112
- [39] Werner M Seiler. *Involution: The formal theory of differential equations and its applications in computer algebra*, volume 24 of *Algorithms and Computation in Mathematics*. Springer, 2010. 29, 112

- [40] Yoshiyuki Sekiguchi and Hayato Waki. Perturbation analysis of singular semidefinite and its application to a control problem. *Preprint arXiv:1607.055681*, 2016. 106, 108
- [41] A.J. Sommese and C.W. Wampler. *The Numerical solution of systems of polynomials arising in engineering and science*, volume 99. World Scientific, 2005. 1, 4, 25, 53, 93
- [42] F. Sottile. *Real solutions to equations from geometry*, volume 57 of *University Lecture Series*. American Mathematical Society, Providence, RI, 2011. 8, 53, 54, 55, 56, 84, 93, 94, 120
- [43] Hans J. Stetter. *Numerical polynomial algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2004. 29, 57, 84, 85, 120, 129
- [44] G.W. Stewart. On the continuity of the generalized inverse. *SIAM J. Appl. Math.*, 17:33–45, 1969. 106
- [45] J.F. Sturm. Error bounds for linear matrix inequalities. *SIAM J. Optim.*, 10(4):1228–1248 (electronic), 2000. 102
- [46] A.D. Wittkopf and G.J. Reid. Fast differential elimination in c: The cdiffelim environment. *Computer Physics Communications*, 139(2):192–217, 2001. 24, 31, 33, 59, 112, 128
- [47] H. Wolkowicz and Q. Zhao. Semidefinite programming relaxations for the graph partitioning problem. *Discrete Appl. Math.*, 96/97:461–479, 1999. Selected for the special Editors’ Choice, Edition 1999. 54, 94
- [48] W. Wu and G.J. Reid. Finding points on real solution components and applications to differential polynomial systems. In *Proceedings of the 38th international symposium on International symposium on symbolic and algebraic computation*, pages 339–346. ACM, 2013. 5, 26, 83, 85, 121
- [49] X. Wu and L. Zhi. Determining singular solutions of polynomial systems via symbolic–numeric reduction to geometric involutive forms. *Journal of Symbolic Computation*, 47(3):227–238, 2012. 24, 59, 112

- [50] Q. Zhao, S.E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *J. Comb. Optim.*, 2(1):71–109, 1998.
- Semidefinite programming and interior-point approaches for combinatorial optimization problems (Fields Institute, Toronto, ON, 1996). 54, 94

# Chapter 5

## Conclusion

Polynomial systems and the need to analyze their real solutions occur frequently in applications. Many methods exist for finding some approximate solutions based on initial guesses sufficiently close to a desired solution. Much less is available for numerically describing aspects of all solutions, especially in the case of real manifolds of solutions. Currently the most promising methods, critical point methods, theoretically find at least one point on each connected real solution component. However, these methods suffer from serious numerical difficulties due to multiplicities, singularities and sums of squares. The main goal of this thesis is to find an equivalent form of the polynomial system, the real radical, which is free of multiplicities and sums of squares.

Our work to numerically determine the real radical was also motivated by the breakthroughs by Lasserre et al [5, 6]. They showed that the real radical could be numerically determined by reformulation as a maximum-rank SDP problem, with a rank stabilization criterion in the 0-dimensional case. Further they improved their 0-dimensional approach by using a prolongation-projection method based on the approach by using the geometric involutive form (GIF) of Wittkopf and Reid [9]. Ma, Wang and Zhi [7] conjectured an extension to positive dimension of determination of real radical by using Pommaret-involutive bases, coupled with an interior point solver.

In chapter 2, an initial exploration is made of an extension to positive dimension using the GIF coupled with the interior point solver SeDuMi. A method is given for extracting lower degree GIF. Reduction of degree techniques are critical and have been extensively developed

in the symbolic case for Gröbner bases [4] and triangular decompositions [2, 3]. GIF are orthogonal bases, found using stable SVD techniques, unlike non-orthogonal Pommoret Bases used in Ma, Wang and Zhi [7]. In chapter 2, we gave a stopping criterion for computing an intermediate basis between a polynomial system and its real radical.

The work [6] motivated us to combine SDP – moment matrix methods with our geometric involutive bases to approximate positive dimensional real radical ideals. In particular, the termination criterion  $\text{rank}(\mathbf{M}(Q)) = \dim \ker \text{GIF}(Q)$  in Algorithm 2.6.1 is equivalent to the rank stabilization condition in Lasserre [6] for zero dimensional systems.

The approach of chapter 3 is motivated by geometrical and accuracy issues. In particular, geometrically the generic point that is computed in our SDP Moment matrix approach lies at the intersection of a cone of semi-definite matrix and an affine space tangent to the cone, i.e., at a face of the cone. Arbitrarily small perturbations move the generic point to infeasible region with associated numerical difficulties. To address the difficulty Borwein and Wolkowicz introduced “facial reduction”. Working with Wolkowicz, in Chapter 3, we introduced facial reduction for our SDP problems. Facial reduction yields an equivalent problem for which there are strictly feasible points on the interior of a face. Facial reduction also reduces the size of the moment matrices occurring in the application of SDP methods. For example the determination of a  $k \times k$  moment matrix for a problem with  $m$  linearly independent constraints is reduced to a  $(k - m) \times (k - m)$  moment matrix by one facial reduction. The high accuracy required by facial reduction and also the ill-conditioning commonly encountered in numerical polynomial algebra [8] motivated us to implement Douglas-Rachford iteration. We use facial reduction with our MATLAB implementation of Douglas-Rachford iteration (our FDR method). In the case of only one constraint, say as in the case of univariate polynomials, one might expect that the improvement in convergence due to that facial reduction would be minor. However we present a class of random univariate polynomials, where one such facial reduction combined with DR iteration, yields the real radical much more efficiently than the standard interior point method in SeDuMi.

In chapter 4, we studied cases with more than 1 facial reduction and proved the maximum rank property is attained by our method. We gave an algorithm to compute the generators of real radical ideal up to a given degree.

We established an algorithm which can compute the maximum rank solution using primal and dual form. We gave the first examples which involve more than 1 facial reductions (singularity degree more than 1). We showed in the examples that the maximum rank solution can be computed accurately even if the singularity degree is more than 1. Our algorithm based on DR iteration was much more accurate than the interior point solver SeDuMi on our test examples.

In addition we discussed the well-posedness of facial reduction. We showed the maximum rank doesn't change under sufficiently small perturbations.

Highlights of this thesis:

(0) In comparison to previous work, the thesis gives a much deeper exploration of the underlying numerics and geometry of SDP-Moment matrix techniques for polynomial systems.

(1) We gave an improved geometric involutive bases algorithm (GIF) which involves projection to lower degree equivalent systems reducing the cost.

(2) Combining with facial reduction and powerful Douglas-Rachford projection-reflection method, we are able to compute the maximum rank moment matrix with much higher accuracy and in a more stable way than the classical interior point solver SeDuMi. Our examples show that facial reduction is essential in order to get accurate and reliable results especially for examples with singularity degree more than 1.

(3) Compared with the "local" method to compute the real radical ideal, we give a stable global method to compute the generators of real radical ideal up to any given degree. This also yields a solution of the real radical membership problem. Previous approximate real radical membership algorithms don't have a degree bound, so no guarantee for termination in finite many steps. Combined with the recent work by Hauenstein et al [1], one can have a complete algorithm for computing the real radical ideal in positive dimension which terminates in finitely many steps.

(4) This thesis further contributes to recent remarkable connections between Algebraic Geometry (an area with relatively few researchers), and convex optimization (a vast area with many practitioners).

Future work:

(1) We are planning to do a more thorough analysis for the perturbation of facial reduction algorithm.

(2) We are also planning to develop a better critical point method approach to be combined with the approach described in this thesis to compute the real radical ideal in positive dimension case.

(3) The widespread applications of real radicals in Science, Engineering and Mathematics motivate the development of user-friendly implementations of the algorithms of the thesis.

## Bibliography

- [1] D. Brake, J. Hauenstein, and A. Liddell. Numerically validating the completeness of the real solution set of a system of polynomial equations. *Proceedings of the 41th International Symposium on Symbolic and Algebraic Computation*, 2016. 1, 115, 118, 121, 130
- [2] Changbo Chen, James H Davenport, John P May, Marc Moreno Maza, Bican Xia, and Rong Xiao. Triangular decomposition of semi-algebraic systems. In *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, pages 187–194. ACM, 2010. 25, 46, 129
- [3] Changbo Chen, Marc Moreno Maza, Bican Xia, and Lu Yang. Computing cylindrical algebraic decomposition via triangular decomposition. In *Proceedings of the 2009 international symposium on Symbolic and algebraic computation*, pages 95–102. ACM, 2009. 25, 46, 129
- [4] Jean-Charles Faugere. A new efficient algorithm for computing gröbner bases. *Journal of pure and applied algebra*, 139(1):61–88, 1999. 46, 129
- [5] Jean Bernard Lasserre, Monique Laurent, and Philipp Rostalski. Semidefinite characterization and computation of zero-dimensional real radical ideals. *Foundations of Computational Mathematics*, 8(5):607–647, 2008. 13, 108, 110, 128
- [6] M. Laurent and P. Rostalski. The approach of moments for polynomial equations. In Miguel F. Anjos and Jean B. Lasserre, editors, *Handbook on semidefinite, conic and polynomial optimization*, International Series in Operations Research & Management Science,

- 166, pages 25–60. Springer, New York, 2012. 10, 13, 14, 15, 16, 17, 36, 38, 40, 46, 47, 58, 95, 96, 128, 129
- [7] Y. Ma, C. Wang, and L. Zhi. A certificate for semidefinite relaxations in computing positive dimensional real varieties. *Journal of Symbolic Computation*, 72:1 – 20, 2016. vii, x, 1, 8, 53, 55, 62, 77, 78, 79, 80, 81, 82, 84, 93, 120, 121, 128, 129
- [8] Hans J. Stetter. *Numerical polynomial algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2004. 29, 57, 84, 85, 120, 129
- [9] A.D. Wittkopf and G.J. Reid. Fast differential elimination in c: The cdiffelim environment. *Computer Physics Communications*, 139(2):192–217, 2001. 24, 31, 33, 59, 112, 128



# Appendix A

## Proof of Primal Theorem of Alternative

**Theorem A.0.1 (Primal Theorem of alternative)** Suppose  $\mathcal{A} : \mathcal{S}_+^k \rightarrow \mathbb{R}^m$  is a linear transformation,  $b \in \mathbb{R}^m$ ,  $P \in \mathcal{S}^k$  and  $Z \in \mathcal{S}^k$ . Then exactly one of the following alternative systems is consistent.

$$(I) \quad 0 < P \in F := \{P \in \mathcal{S}^k : \mathcal{A}(P) = b, P \geq 0\} \quad (\text{A.1a})$$

$$(II) \quad 0 \neq Z = \mathcal{A}^*y \geq 0, b^T y = 0. \quad (\text{A.1b})$$

**Proof  $\implies$ :** Assume (A.1a) holds and the left hand side of (A.1b) holds, then

$$0 = b^T y = \langle \mathcal{A}(P), y \rangle \quad (\text{A.2})$$

$$= \sum_{i=1}^n \text{trace}(A_i P) y_i = \sum_{i=1}^n \text{trace}(A_i y_i P) \quad (\text{A.3})$$

$$= \text{trace}\left(\sum_{i=1}^n (A_i y_i) P\right) = \text{trace}(ZP) \quad (\text{A.4})$$

$$= \langle P, Z \rangle. \quad (\text{A.5})$$

Then  $\langle P, Z \rangle = 0$  implies  $PZ = 0$ . So  $\text{range } P \subseteq \text{null } Z$ . Therefore, if  $P > 0$ , then  $\text{range } P = \mathbb{R}^n$  and  $\text{null } Z = \mathbb{R}^n$ , so  $Z = 0$ .

$\Leftarrow$ : To show (A.1b) implies (A.1a), we define:

$$\bar{A}_i = \begin{pmatrix} -b_i & 0 \\ 0 & A_i \end{pmatrix}, \quad \bar{X} \in \mathcal{S}^{k+1}, \quad \bar{Y} \in \mathcal{S}^{k+1}.$$

Then (A.1b) is equivalent to saying  $\bar{Z} = \bar{\mathcal{A}}^*y \geq 0 \implies \bar{Z} = 0$ . To see this, suppose  $\bar{Z} = \bar{\mathcal{A}}^*y \geq 0$ , then it implies  $-b^T y \geq 0$ . But weak duality also implies  $-b^T y \leq 0$ , so  $b^T y = 0$ .

So suppose  $\bar{Z} = \bar{\mathcal{A}}^*y \geq 0$  only has zero solution, we want to prove that (A.1a) holds, or equivalently, to prove that  $\exists \bar{X} \in \mathcal{S}^{k+1}, \bar{\mathcal{A}}(\bar{X}) = 0, \bar{X} > 0$ .

Suppose  $y_1 \bar{A}_1 + \cdots + y_n \bar{A}_n \geq 0$  only has zero solution, then the linear subspace  $\mathcal{L} = y_1 \bar{A}_1 + \cdots + y_n \bar{A}_n$  is disjoint from the interior of the cone  $\mathcal{S}_+^{k+1}$ . By the hyperplane separation theorem, there exists a hyperplane containing this linear subspace that is disjoint from the interior of  $\mathcal{S}_+^{k+1}$ . So there exists  $\bar{X}$  such that  $\bar{Y} \cdot \bar{X} = 0$  for  $\bar{Y} \in \mathcal{L}$  and  $\bar{Y} \cdot \bar{X} > 0$  for  $\bar{Y} \neq 0 \in \mathcal{S}_+^{k+1}$ . Note that the top left element of  $\bar{X}$  can't be zero, otherwise  $\bar{Y}$  can be chosen in this way such that only the top left element is one while all the others are zero. So  $\bar{Y} \cdot \bar{X} = 0, \bar{Y} \geq 0, \bar{Y} \neq 0$ , a contradiction. Therefore, we can infer  $\bar{X} > 0$ . Because  $X$  is a principal submatrix of  $\bar{X}$ , we conclude  $X > 0$ . Also  $\bar{\mathcal{A}}(\bar{X}) = 0$  since  $\bar{A}_i \in \mathcal{L}$  which implies  $\mathcal{A}(X) = b$ .

■

# Appendix B

## Copyright Release

- A version of Chapter 2 has been published by *Proceedings of the 2014 Symposium on Symbolic-Numeric Computation*, Pages 41-42. ACM, 2014. In the publisher's website for ACM Author Rights and Publishing Policy under section 2.5, they state:

*The original Owner/Author permanently holds these rights:*

- *Post the Accepted Version of the Work on (1) the Author's home page, (2) the Owner's institutional repository, (3) any repository legally mandated by an agency funding the research on which the Work is based, and (4) any non-commercial repository or aggregation that does not duplicate ACM tables of contents, i.e., whose patterns of links do not substantially duplicate an ACM-copyrighted volume or issue. Non-commercial repositories are here understood as repositories owned by non-profit organizations that do not charge a fee for accessing deposited articles and that do not sell advertising or otherwise profit from serving articles.*
  - *Post an "Author-Izer" link enabling free downloads of the Version of Record in the ACM Digital Library on (1) the Author's home page or (2) the Owner's institutional repository;*
- A version of Chapter 3 has been submitted to *Theoretical Computer Science*, Elsevier. In the publisher's website, under "AUTHOR AND USER RIGHTS", they state:

*As a journal author, you have rights for a large range of use of your article, including*

*use by your employing institute or company. These rights can be exercised without the need to obtain specific permission.*

- A preprint of Chapter 4 has been submitted to arXiv:1606.00491. The author can post this preprint on the owner's institution for free download without any permission from arXiv.

# Curriculum Vitae

**Name:** Fei Wang

**Post-Secondary** University of Western Ontario

**Education and** London, Ontario, Canada

**Degrees:** 20012 - 2016 Ph.D.

University of Waterloo

Waterloo, Canada

2010- 2011 M.S.

University of Science & Technology

Beijing, China

2006 - 2011 B.A.

**Related Work** Teaching Assistant

**Experience:** The University of Western Ontario

2012 - 2016

## Publications:

- Greg Reid, Fei Wang and Wenyan Wu. A note on geometric involutive bases for positive dimensional polynomial ideals and SDP methods. *Proceedings of the 2014 Symposium on Symbolic-Numeric Computation*, Pages 41-42. ACM, 2014.

- Robert M. Corless, David J. Jeffrey and Fei Wang. The asymptotic analysis of some interpolated nonlinear recurrence relations. *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, Pages 115-121, ACM, 2014.
- Ming-Wen Chen, Xin-Feng Wang, Fei Wang and Guo-Biao Lin and Zi-Zong Wang. The effect of interfacial kinetics on the morphological stability of a spherical particle. *Journal of Crystal Growth*, Pages 20-23, Elsevier, 2013.
- Fei Wang. Proof of a series solution for Euler's trinomial equation , submitted to *ACM Communications in Computer Algebra*, 2016.
- Greg Reid, Fei Wang, Henry Wolkowicz and Wenyuan Wu. Facial Reduction and SDP Methods for Systems of Polynomial Equations. submitted to *Theoretical Computer Science*, arXiv:1504.00931, 2015.